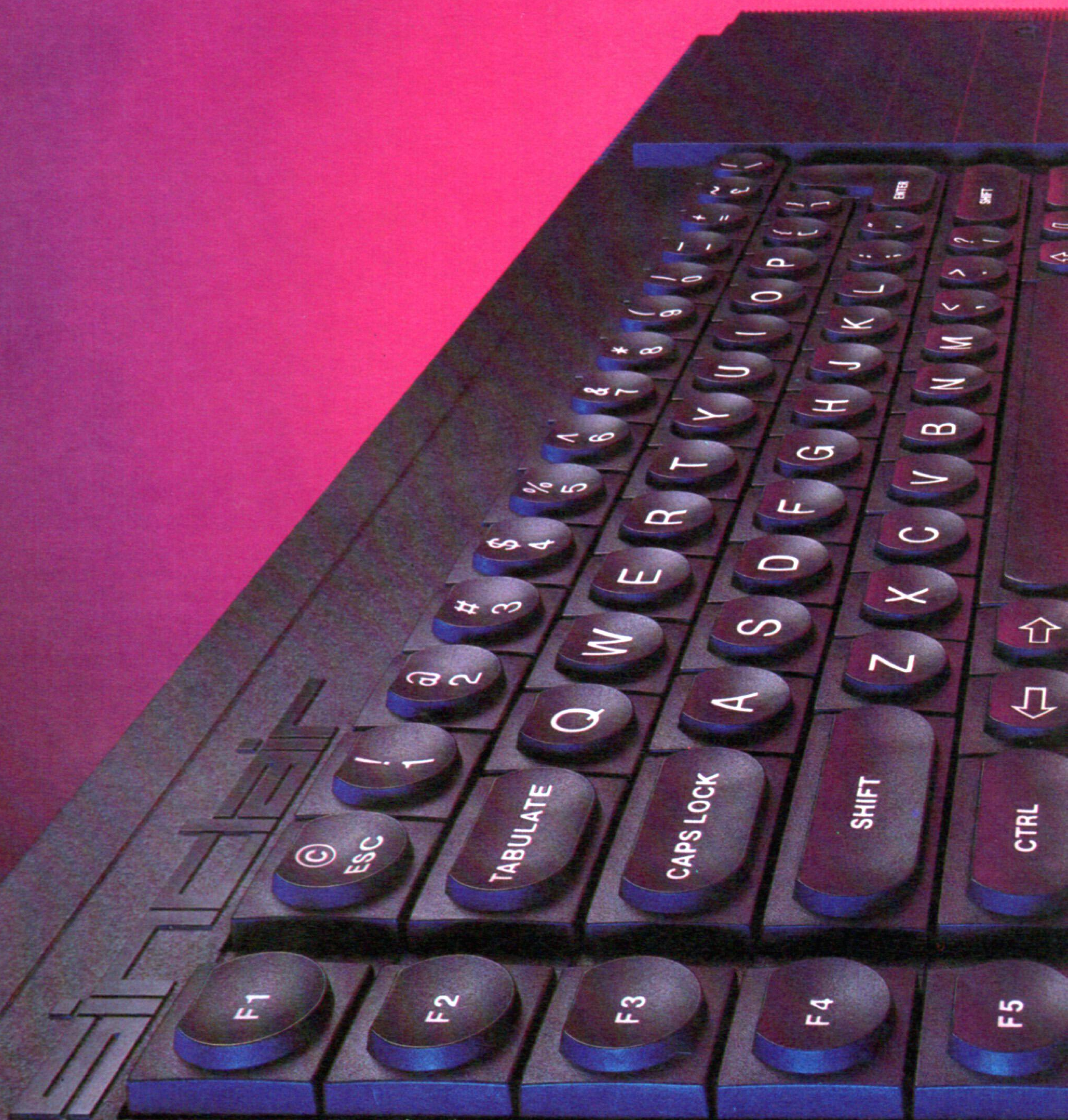
150 ptas.

mi computer

50

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR



Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona

Volumen V - Fascículo 50

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones, Francisco Martín
Asesor técnico: Ramón Cervelló

Redactores y colaboradores: G. Jefferson, R. Ford, F. Martín, S. Tarditti, A. Cuevas, F. Blasco
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, 08008 Barcelona
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador*, se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-7598-007-4 (tomo 5)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 268412
Impreso en España - Printed in Spain - Diciembre 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, 28034 Madrid.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Edificio Bloque Dearmas, final Avda. San Martín con final Avda. La Paz, Caracas 1010.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)
Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 6.850.277 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Editorial Delta, S.A. (Paseo de Gracia, 88, 5.°, 08008 Barcelona), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Editorial Delta, S.A., en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.

Editorial Delta, S.A.



Un paso más allá

En este artículo se pretende arrojar un poco de luz sobre el controvertido ordenador QL de Sinclair



Ian McKinnell

La familia reunida
El nombre Sinclair ha aparecido en una considerable gama de productos: desde amplificadores de audio al celebrado Black Watch, pasando por calculadoras y ordenadores, hasta el tanto tiempo esperado televisor de pantalla plana y el coche eléctrico. Habilidad técnica e innovación, diseño de producto de alta tecnología con estilo y ambiciosas estrategias de marketing han sido los distintivos de Sinclair desde el principio, si bien sus detractores afirmarían que las etiquetas más ajustadas a la realidad serían ingeniería de comunicados de prensa, artimañas y plazos de entrega excesivamente optimistas

No es sorprendente que haya tanta gente confundida sobre el Sinclair QL y su verdadera valía. Primero fue la expectación que despertaron los medios de comunicación a raíz de su lanzamiento. Se le dijo al público que el QL era el ordenador personal más avanzado que existía hasta la fecha, con una CPU 68008 de 32 bits, un juego de cuatro programas de software de gestión integrado y un BASIC residente que superaba a cualquiera de las versiones existentes.

Sin embargo, esta expectación se fue desvaneciendo a medida que las fechas de entrega se retrasaban una y otra vez, y el público comprendió que quizá había ligeras exageraciones en lo que se afirmaba del QL. Se llegó a un auténtico punto muerto cuando quedó en evidencia (a pesar de todas las seguridades dadas por Sinclair) que la máquina no estaba ni por asomo lista para su venta al público: todo el dinero que habían adelantado los clientes mediante pedidos por correo estaba simplemente depositado en las cuentas bancarias de Sinclair produciendo intereses, mientras el ansioso público esperaba y recordaba anteriores lanzamientos de máquinas Sinclair y su entrega.

Entre el lanzamiento del Spectrum y el del QL había, no obstante, una gran diferencia; a pesar de que no se cumplieron los plazos de entrega de 28 días del Spectrum, poco después del lanzamiento se proporcionaron modelos a los periodistas para que los sometieran a examen. Las máquinas funcionaban y eran virtualmente idénticas a las que luego salieron a la venta. En el caso del QL, sin embargo, Sinclair ha tenido que admitir que el "SuperBASIC" y el sistema operativo prometidos no cabían en los 32 Kbytes de ROM asignados: se hubieran necesitado 48 Kbytes, ¡pero en la placa de circuito impreso no había lugar para el chip extra!

En vez de perder tiempo y dinero volviendo a diseñar la placa, Sinclair lanzó su nada afortunado invento denominado *kludge* (llamado algunas veces, equivocadamente, "dongle"). Era una pequeña caja de plástico negro que sobresalía de la puerta para cartuchos del QL y contenía las partes del BASIC y del sistema operativo que no cabían en los 32 K.

Esto al menos le permitió a Sinclair sacar de la fábrica algunas máquinas en condiciones de funcionamiento, con la promesa de que se les quitaría el kludge más adelante, cuando saliera la verdadera máquina mejorada. Los 28 días originales de plazo de entrega se convirtieron en tres meses. Se produjeron en rápida sucesión varias versiones del sistema operativo, cada una de ellas con sus propios defectos. Finalmente, Sinclair se decidió por uno llamado "AH", y éste se ha convertido en la primera versión que sale a la venta en grandes cantidades.

La reacción del consumidor ante el kludge no fue

favorable: era una prueba visible de la insuficiencia de la máquina, y no constitutía ningún indicio de fiabilidad. Para sortear esta reacción, Sinclair tuvo una idea que, con toda seguridad, habrá dejado atónitos a los ingenieros electrónicos: dado que en la placa de circuito impreso no había ningún conector para el tercer chip de 16 Kbytes, éste se colocó "sobre los hombros" del chip existente, y las 28 patas del chip, excepto una, se soldaron individualmente a las del chip de abajo. La última pata se conectó mediante un cable flotante a otra parte del tablero, de modo que se pudiera acceder al nuevo chip independientemente de su anfitrión. Esto supuso la eliminación del kludge, pero en realidad la situación no se había modificado.

Todas las primeras máquinas utilizaban unidades EPROM en vez de ROM, ahorrándole a Sinclair el tiempo que lleva producir unidades de ROM, pero costándole dinero a la empresa. Este alto dispendio posiblemente haya obligado a Sinclair a una temprana aceptación de la versión AH de sistema operativo, a pesar de sus diversos errores. Después de haber arreglado la situación de forma definitiva, se podían fabricar e instalar las ROM para sustituir a las costosas EPROM y la empresa tuvo la esperanza de comenzar a obtener beneficios con la máquina. Lamentablemente, es poco probable que la versión depurada del AH aparezca de inmediato.

Teniendo presente todos estos detalles, ¿cómo podemos juzgar al QL? Se lo puede considerar como dos máquinas en una: un potente ordenador personal o una modesta máquina de oficina y, como tal, se la puede calificar de auténtica pionera. Es probable, sin embargo, que esté más cerca de la idea convencional de lo que es un micro personal: es pequeño, puede producir una visualización en televisión, tiene un BASIC residente, se vende en base a pedidos por correspondencia (y pronto también directamente en tiendas comerciales) y posee gráficos a color en alta resolución y puerta para palanca de mando. Posee dos características que le confieren el estatus de máquina de oficina: los microdrives incorporados proporcionan un sustancial almacenamiento masivo (en comparación con las cassettes) y la máquina viene "arropada" con cuatro programas de aplicaciones: tratamiento de textos, hoja electrónica, base de datos y soporte para gráficos.

Como micro personal el QL parece una muy buena alternativa, dados los microdrives y el software, sobre todo si se considera la alta calidad de éste y el hecho de que los usuarios de ordenadores personales en raras ocasiones ven bases de datos u hojas electrónicas. Por otra parte, la mayoría de ellos no necesitan estas aplicaciones y no tienen ninguna utilización que darles. Al típico usuario de ordenador personal le agrada entretenerse con juegos y escribir programas en BASIC, lo que resulta muy atractivo en este último caso, ya que el SuperBASIC es ciertamente una de las mejores versiones que se han producido hasta la fecha. Existe una escasez obvia de software comercial para el QL, debido a los problemas que se plantearon en la producción de la máquina. La producción de software no se ve facilitada por la incompatibilidad de los nuevos microdrives con la versión para el Spectrum, por el hecho de que la pantalla del QL tenga un trazado distinto, y por las diferencias y los problemas relativos al sistema operativo.

Rivalidad

QL contra BBC Micro
Para aspirar al mercado de usuarios personales serios, el QL debe competir con el BBC Micro. A favor del QL, el BBC Micro no incluye software gratuito, almacenamiento masivo, un microprocesador moderno, gráficos artísticos ni una gran memoria ampliable; por otra parte, el QL no posee la aprobación del gobierno británico para su utilización en las escuelas, ni una enorme base de usuarios en hogares, escuelas y empresas, una inmensa gama de periféricos de alta calidad producidos por terceros, un extenso catálogo de software ni capacidad de ampliación mediante un segundo procesador



QL contra Macintosh
Sir Clive Sinclair fijó la fecha de introducción del QL como para hacerle sombra a la inmitente introducción del Macintosh. Al tomar esta decisión, y al diseñar una máquina basada en el mismo microprocesador, Sinclair ha colocado al QL en una posición de directa rivalidad con el Mac, una máquina de precio cuatro o cinco veces superior. A pesar de que las dos máquinas comparten muchas características, incluyendo el tamaño de memoria, la velocidad del reloj de la CPU y el software integrado, en realidad entre el QL y el Macintosh no cabe ninguna comparación. El QL es una máquina técnicamente brillante, pero no hay nada nuevo en la forma en que opera. El Macintosh de Apple, por otra parte, posee un sistema operativo con iconos, ventanas y el ingenioso "ratón", que coloca a la máquina en un lugar muy alejado de todos los otros ordenadores y la convierte en precursora de una nueva generación de micros



Aparentemente en el SuperBASIC aún persisten algunos errores, y el editor es muy similar al editor de líneas utilizado en el Spectrum; esto no constituye un error, por supuesto, pero tampoco responde exactamente al estándar de la quinta generación. La adición de procedimientos y funciones al estilo BBC y de una estructura SELECT similar a la CASE del PASCAL son mejoras muy reales, si bien no se ofrece ninguna facilidad del tipo de la instrucción ON ERROR. Los gráficos son muy buenos y el sonido, aunque decepcionante, al menos es audible.

Como máquina de oficina el QL es menos convincente: el software que lo acompaña vale la pena, pero seguramente cualquier usuario de gestión de-

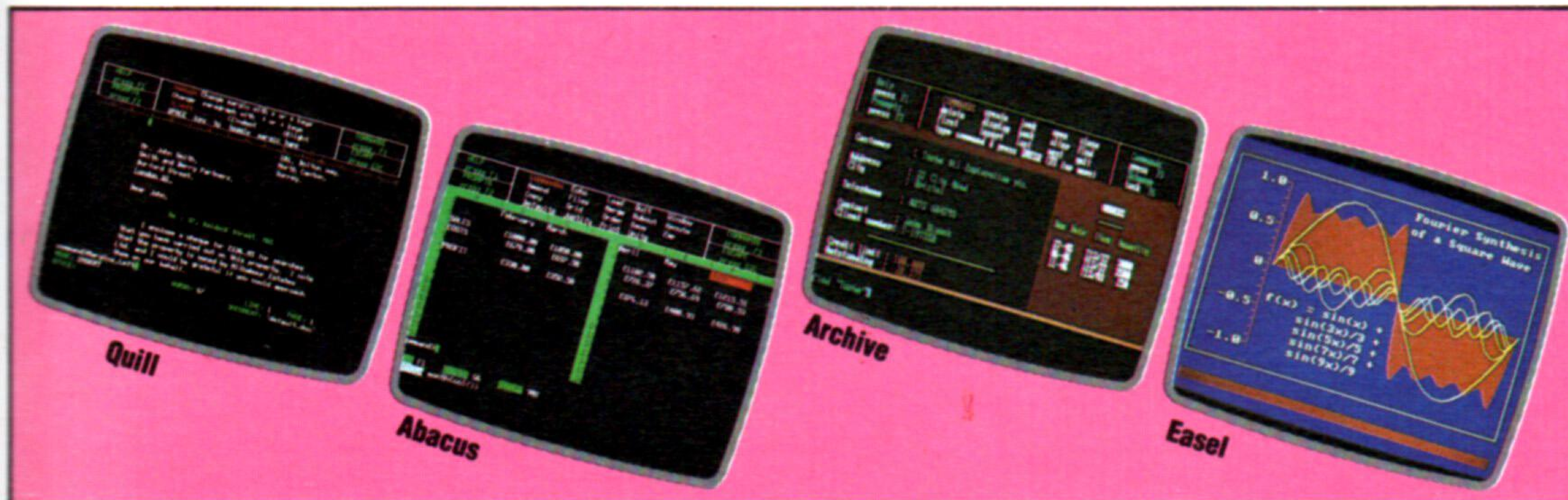


Software integrado
Los cuatro paquetes poseen formatos de pantalla similares e instrucciones y visualización claras y concisas. Se pueden trasladar datos entre ellos mediante el microdrive. Los cuatro programas tienen errores en su versión inicial, como instrucciones que no funcionan y errores de entrada-salida; pero éstos resultarán fáciles de solventar en versiones posteriores. Quill, un procesador de textos, permite visualizaciones de 40, 64 u 80 caracteres; los caracteres digitados tardan en aparecer en la pantalla, lo que puede resultar irritante. Abacus es una innovadora hoja electrónica con muchas funciones incorporadas y la capacidad de etiquetar y direccionar un grupo de celdas, pero el espacio de trabajo sobrante, de 15 K, es virtualmente inaprovechable. Archive es una base de datos con instrucciones incorporadas para tareas de archivo sencillas. También se puede programar mediante un lenguaje interno similar al SuperBASIC, pero el microdrive hace que resulte muy lento. Easel crea gráficos de barras, diagramas tipo "tarta" y gráficos lineares de datos numéricos, y se puede pasar rápidamente de un formato a otro





Ian McKinnell



Ian McKinnell

SINCLAIR QL

DIMENSIONES
472 × 138 × 46 mm

CPU
Motorola 68008, 7,5 MHz

MEMORIA
128 K de RAM (ampliables a 640 K); 48 K de ROM

PANTALLA
25 líneas de 80 caracteres (con pantalla); gráficos en alta resolución: 512 × 256 pixels (4 colores), 256 × 256 (8 colores)

INTERFACES
RS232 en serie (2), palancas de mando (2), microdrives, LAN, TV, pantalla RGB

LENGUAJES DISPONIBLES
SuperBASIC

TECLADO
Pseudoestilo máquina de escribir; 65 teclas, incluyendo auténtica barra espaciadora y cinco teclas de función, pero sin tecla de borrado (*delete*)

DOCUMENTACION
El manual del usuario responde a un estándar elevado e incluye manuales para SuperBASIC y el software de aplicaciones

VENTAJAS
CPU 68008 de proceso de números muy veloz, software de gran calidad incluido, gráficos muy buenos, versión avanzada de BASIC

DESVENTAJAS
Los microdrives incorporados son más lentos que los discos y no son compatibles con el Spectrum; hay poco software disponible y el sistema operativo no está depurado por completo

seará al menos un paquete de contabilidad, y el programa de hoja electrónica deja sólo 15 Kbytes de RAM para el usuario, lo que descarta a la mayoría de modelos financieros serios. La velocidad y la cuestionable fiabilidad de los microdrives pone en entredicho toda la capacidad de almacenamiento masivo del QL, especialmente al no haber ninguna interface para unidad de disco. El teclado no tiene aspecto de resistir un uso diario intensivo y resulta difícil imaginar que los mecanógrafos cualificados acepten sus peculiaridades. La escasez de software comercial constituye un inconveniente más grave para quienes empleen la máquina con fines de gestión, y esto, unido a las deficiencias del almacenamiento masivo, es probable que dé por concluida la carrera del QL antes de que siquiera comience.

Al igual que todos los productos Sinclair, el QL es atrayente, innovador y objeto de controversia. Aunque en realidad no se puede decir que haya cubierto ninguno de sus objetivos, el QL ha proporcionado a sus competidores un nuevo estándar y un punto de referencia para establecer comparaciones.



Memoria burbuja

El QL posee un teclado tipo "membrana". Dos cables de señal se mantienen separados mediante una burbuja formada en una membrana plástica; cuando se pulsa la tecla, ésta aplasta la burbuja y los cables entran en contacto. La resistencia y la fuerza de retorno las proporciona la "ampolla" plástica. Con teclas plásticas esculpidas de recorrido total, el teclado del QL supone una enorme mejora respecto al del Spectrum, pero su construcción endeble y su tacto esponjoso lo colocan en una situación de desventaja en relación al Vic-20

Rosalind Buckland

Amenaza mortal

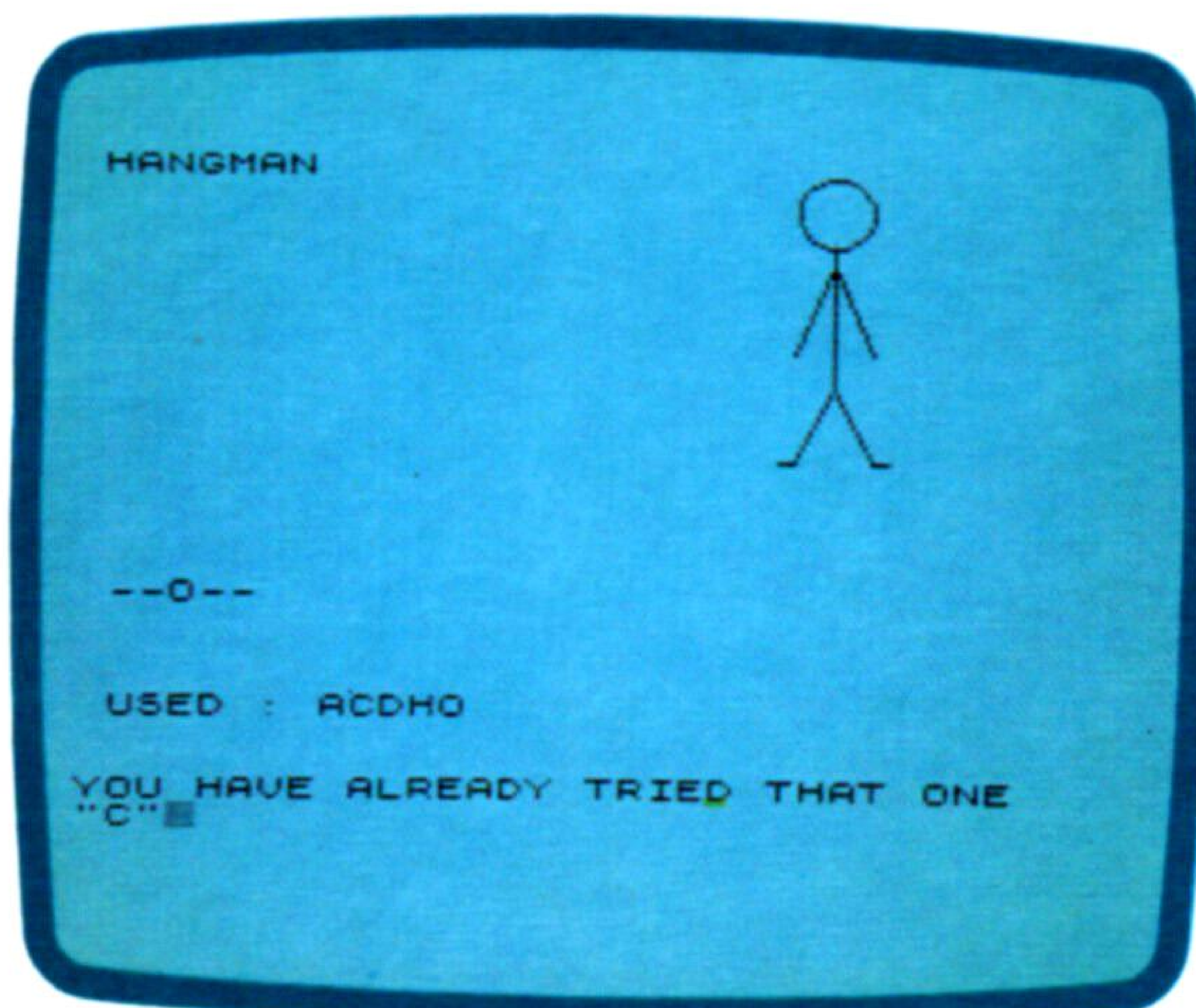
El "Ahorcado" es un tradicional juego de palabras cuya implementación para ordenador es sencilla y resulta de gran valor educativo

Probablemente todos hayamos participado en algún momento en el juego del *Ahorcado*. El objetivo del juego consiste simplemente en averiguar las letras de una palabra. La única información que se proporciona es la cantidad de caracteres que posee la palabra, todos los cuales se representan mediante guiones. Una letra adivinada con acierto se visualiza en su posición correcta, y una conjetura incorrecta hace que se dibuje uno de los elementos de la imagen de un hombre sobre un patíbulo. En nuestro programa este dibujo se compone de 10 partes, que se muestran sobre el lado derecho de la pantalla. Si se completan las 10 partes antes de haber completado la palabra entera, entonces el hombre es ahorcado y uno pierde el juego.

El principio básico de nuestro programa es muy simple. Implica verificar si una letra entrada por teclado forma parte de una palabra "secreta" seleccionada al azar (una de once palabras contenidas en sentencias DATA al final del programa). Si la letra aventurada por el jugador está presente, se visualiza en el lugar correcto dentro de la palabra. Si la letra no está presente, el programa debe visualizarla de todas maneras para recordarle al jugador que la misma ya ha sido empleada. Además, el programa debe entonces saltar a una subrutina que dibujará un elemento del hombre colgado.

Las palabras que utiliza nuestro programa están almacenadas, en ambas versiones, en las líneas 1020 y 1030. No existe ninguna razón para no agregar otras, valiéndose de más sentencias DATA. Pero si usted agrega su propio léxico rompedor de cabezas, debe recordar que su longitud no debe superar las diez letras. (Si bien esta limitación se podría ampliar modificando las líneas 30 y 50.) Asimismo, se debe agregar el número total de palabras de las sentencias DATA y alterar consiguientemente el valor N de la línea 20.

Kevin Jones



Liz Heaney

Informe de la situación
El juego en plena ejecución, mostrando el desarrollo del hombre colgado y la evolución de la palabra

Al comienzo del programa todas las palabras se leen sobre una matriz. Se escoge al azar uno de los elementos de la matriz y se visualiza en la pantalla una línea de guiones correspondiente a la longitud de la palabra. El resto del juego consiste en un bucle repetitivo. Cuando se entra cualquier cosa por teclado, lo primero es verificarla: si se trata de más de una letra, o de ningún carácter en absoluto, entonces el programa hace un BEEP y vuelve a repetir el bucle en busca de otra entrada. La letra también se verifica con la lista de letras que ya han salido en el juego. Si ya se la ha utilizado antes, entonces aparece en la pantalla un mensaje intermitente y se solicita una nueva letra.

Si la letra aventurada es aceptable, se la agrega a la lista visualizada de caracteres utilizados y luego se la compara con cada una de las letras de la palabra, de una en una. Si concuerda con cualquier posición, se la coloca en la pantalla en lugar del guión correspondiente. Si no concuerda con ninguna letra de la palabra, se llama a una subrutina que dibuja una de las partes del ahorcado.

Cuando el jugador ha hecho 10 conjeturas erróneas, el hombre es ahorcado, se toca una breve melodía de consolación y se selecciona una nueva palabra. De lo contrario, cuando se han colocado correctamente todas las letras de la palabra, se toca una frase musical de felicitación.

Nuestras dos versiones del programa se pueden adaptar fácilmente para la mayoría de los micros personales. Por supuesto, es necesario adaptar las subrutinas que dibujan el hombre y el patíbulo especialmente, de acuerdo a las capacidades para gráficos de cada máquina determinada. Tal vez a los programadores interesados en crear interesantes visualizaciones en pantalla les agrade elaborar más el dibujo final: ¿un hombre ahorcado, colgando suavemente empujado por la brisa cuando el jugador pierde, quizá?

Al juego también se le pueden agregar otros refinamientos. A modo de sugerencia, sería una buena idea agregar una comprobación al principio del programa para ver si todavía no se ha utilizado alguna de las palabras seleccionadas. Tal como está el programa, la selección de éstas es una cuestión de puro azar, y se podría elegir la misma palabra dos veces una inmediatamente después de la otra.

Aún sería más útil una rutina para comprobar las entradas de modo que sólo se aceptaran las letras en mayúsculas. Las letras minúsculas, los números y los símbolos se podrían comprobar mediante sus códigos ASCII. No obstante, tal como lo ofrecemos aquí, nuestro programa es una buena versión del *Ahorcado* y ofrece oportunidades para los programadores audaces.



El caso en cuestión
Observe que debe solicitarle al jugador del "Ahorcado" que utilice mayúsculas o bien minúsculas, según sea el caso de la palabra misteriosa

BBC Micro

```
10 REM          Ahorcado
20 N=11: REM NUMERO DE PALABRAS EN "DATA"
30 DIM W$(N)
40 REM inicializar
50 D$="__________": REM TANTOS GUIONES COMO LONG. MAX.
60 FOR I=1 TO N
70    READ W$
80    W$(I)=W$
90    NEXT I
100 MODE 1
110 REM comenzar una palabra
120 CLS
130 PRINT TAB(5,1);"AHORCADO"
140 R$=W$(RND(N))
150 C=0:W=0:D=0
160 G$=""
170 PRINT TAB(5,23);"Letras probadas :"
180 L=LEN(W$)
190 PRINT TAB(5,20);LEFT$(D$,L)
200 REM entrar la letra aventurada
210 PRINT TAB(5,30);STRING$(34," ")
220 INPUT TAB(5,30);L$
230 REM verificar letra introducida
240 IF L$="" THEN SOUND 1,-15,50,5:GOTO 210
250 IF LEN(L$) > 1 THEN SOUND 1,-15,50,5:GOTO 210
260 F=0
270 FOR I=1 TO LEN(G$)
280    IF L$ <> MID$(G$,I,1) THEN GOTO 330
290    PRINT TAB(5,30)"ESA LETRA YA LA HAS PROBADO"
300    SOUND 1,-15,50,5
310    T%=TIME:REPEAT UNTIL TIME > T% + 50
320    F=1
330    NEXT I
340 IF F=1 THEN 210
350 G$=G$ + L$
360 PRINT TAB(5,25);G$
370 REM cotejar letra con palabra
380 F=0
390 FOR I=1 TO L
400    IF L$=MID$(W$,I,1) THEN GOSUB 480
410    NEXT I
420 IF F <> 1 THEN GOSUB 670
430 IF D=1 THEN GOSUB 590:GOTO 120
440 IF C=L THEN GOSUB 530:GOTO 120
450 GOTO 210
460 END
470 REM letra coincide
480 PRINTTAB(4 + I,20);L$
490 C=C + 1
500 F=1
510 SOUND 1,-15,200,5:SOUND 1,0,0,2
520 RETURN
530 REM exito!
540 FOR I=1 TO 200 STEP 10
550    SOUND 1,-15,I,2
560    NEXT I
570 RETURN
580 REM fracaso!
590 FOR I=200 TO 1 STEP -10
600    SOUND 1,-15,I,2
610    NEXT I
620 CLS
630 PRINT TAB(5,20);"LA PALABRA ERA : ";W$
640 T%=TIME:REPEAT UNTIL TIME > T% + 200
650 RETURN
660 REM la letra no concuerda
670 W=W + 1
680 ON W GOTO 690,760,780,810,830,870,890,910,930,960
690 VDU29,800;800;
700 MOVE 50,0
710 FOR A=0 TO 7 STEP 0.4
720    DRAW 50*COS(A),50*SIN(A)
730    NEXT A
740 VDU29,0;0;
750 RETURN
760 MOVE 800,750: DRAW 800,550
770 RETURN
780 MOVE 740,450:DRAW 750,450:DRAW 800,550
790 DRAW 850,450:DRAW 860,450
800 RETURN
810 MOVE 750,630:DRAW 800,730:DRAW 850,630
820 RETURN
830 PLOT69,800,800
840 PLOT69,820,820
850 PLOT69,780,820
860 RETURN
870 MOVE 780,790:DRAW 800,770:DRAW 820,790
880 RETURN
890 MOVE 600,400:DRAW 1100,400
900 RETURN
910 MOVE 1000,400:DRAW 1000,900
920 RETURN
930 MOVE 1000,900:DRAW 800,900
940 MOVE 1000,850:DRAW 950,900
950 RETURN
960 MOVE 800,900:DRAW 800,850
970 MOVE 780,790:PLOT 14,800,770:PLOT 6,820,790
980 MOVE 780,770:DRAW 800,780:DRAW 820,770
900 D=1
1000 RETURN
1010 REM las palabras
1020 DATA "MUTACION","REEMBOLSAR","SERAFIN","AHORCADO","BBC"
1030 DATA "SPECTRUM","EXODO","ELEFANTE","XENOFOBO","ARRUGA","GRANADA"
```

Spectrum

```
10 REM          AHORCADO
20 LET N=11
30 DIM X$(N,10)
35 DIM Y(N)
40 REM inicializar
50 LET D$="_________ _ _"
60 FOR I=1 TO N
70 READ W$
75 LET Y(I)=LEN W$
80 LET X$(I)=W$
90 NEXT I
110 REM empezar una nueva palabra
120 CLS
130 PRINT AT 1,1;"AHORCADO"
140 LET I=(1 + INT (RND*N))
145 LET W$=X$(I)
150 LET C=0:LET W=0:LET D=0
160 LET G$=""
170 PRINT AT 20,1;"UTILIZADAS : "
180 LET L=Y(I)
190 PRINT AT 16,1;D$(1 TO L)
200 REM entrar la letra aventurada
220 INPUT L$
230 REM verificar letra introducida
240 IF L$="" THEN BEEP 0.25,-10:GOT TO 210
250 IF LEN L$ > 1 THEN BEEP 0.25,-10:GO TO 210
260 LET F=0
270 FOR I=1 TO LEN G$
280 IF L$ <> G$(I) THEN GO TO 330
290 PRINT AT 21,1;"YA HAS PROBADO CON ESA"
300 BEEP 0.25,-10
310 PAUSE 25:
320 LET F=1
330 NEXT I
340 IF F=1 THEN GO TO 210
350 LET G$=G$ + L$
360 PRINT AT 20,8;G$
370 REM cotejar letra con palabra
380 LET F=0
390 FOR I=1 TO L
400 IF L$=W$(I) THEN GO SUB 480
410 NEXT I
420 IF F <> 1 THEN GO SUB 670
430 IF D=1 THEN GO SUB 590: GO TO 120
440 IF C=L THEN GO SUB 530: GO TO 120
450 GO TO 210
460 STOP
470 REM letra coincide
480 PRINT AT 16,I;L$
490 LET C=C + 1
500 LET F=1
510 BEEP 0.25, + 16
520 RETURN
530 REM exito!
540 FOR I=-10 TO 10
550 BEEP 0.1,I
560 NEXT I
570 RETURN
580 REM fracaso!
590 FOR I=10 TO -10 STEP -1
600 BEEP 0.1,I
610 NEXT I
620 CLS
630 PRINT AT 10,3;"LA PALABRA ERA : ";W$
640 PAUSE 50
650 RETURN
660 REM la letra no concuerda
670 LET W=W + 1
680 IF W=1 THEN GO TO 690
681 IF W=2 THEN GO TO 760
682 IF W=3 THEN GO TO 780
683 IF W=4 THEN GO TO 810
684 IF W=5 THEN GO TO 830
685 IF W=6 THEN GO TO 870
686 IF W=7 THEN GO TO 890
687 IF W=8 THEN GO TO 910
688 IF W=9 THEN GO TO 930
689 IF W=10 THEN GO TO 960
690 CIRCLE 200,150,10
700 RETURN
760 PLOT 200,140: DRAW 0,-40
770 RETURN
780 DRAW -10,-20: DRAW -4,0
790 PLOT 200,100: DRAW 10,-20: DRAW 4,0
800 RETURN
810 PLOT 190,100: DRAW 10,25: DRAW 10,-25
820 RETURN
830 PLOT 200,150
840 PLOT 196,155
850 PLOT 204,155
860 RETURN
870 PLOT 196,148: DRAW 4,-4: DRAW 4,4
880 RETURN
890 PLOT 170,60: DRAW 80,0
900 RETURN
910 PLOT 240,60: DRAW 0,115
920 RETURN
930 DRAW -40,0
940 PLOT 240,165: DRAW -10,10
950 RETURN
960 PLOT 200,175: DRAW 0,-15
970 OVER 1: PLOT 196,148: DRAW 4,-4: DRAW 4,4: OVER 0
980 PLOT 196,144: DRAW 4,4: DRAW 4,-4
990 LET D=1
1000 RETURN
1010 REM las palabras
1020 DATA "MUTACION","REEMBOLSAR","SERAFIN","AHORCADO","BBC"
1030 DATA "SPECTRUM,"EXODO","ELEFANTE","XENOFOBO","ARRUGA","GRANADA"
```

Hacia el Logo

Comenzamos una serie de artículos sobre el LOGO, un lenguaje diseñado con el pensamiento puesto en la educación

Habiendo analizado con todo detalle la programación en BASIC y en lenguaje máquina, iniciamos nuestro curso sobre otros lenguajes de ordenador populares con una serie de capítulos sobre LOGO. Usted se preguntará por qué hemos elegido este lenguaje como tema de un plan de aprendizaje ampliado. Al fin y al cabo, existen muchos otros lenguajes que llevan a cabo ciertas funciones sumamente bien, y son raras las ocasiones en que los ordenadores personales se suministran con LOGO. No obstante, éste ofrece características muy atractivas para el usuario de un ordenador personal.

En primer lugar, el LOGO es uno de los mejores lenguajes introductorios que existen hoy en día para cualquier ordenador. Por supuesto, si usted ya ha estado programando en BASIC quizá no se sienta muy interesado por conocer un lenguaje de introducción. Pero el LOGO puede servir, incluso para un programador de BASIC ya experimentado, como una excelente introducción a la programación "estructurada" y para el empleo de procedimientos en lugar de sentencias. En segundo lugar, el LOGO está disponible en cartucho o en cassette para la mayoría de los ordenadores personales. De hecho, una de las mejores versiones de LOGO existentes es la escrita para el Spectrum que distribuye Sinclair. Por último, el LOGO es un sistema de aprendizaje muy eficaz. Aunque no es sencillo de dominar, el LOGO es uno de los pocos lenguajes de programación con el que es fácil empezar a trabajar.

El LOGO tiene sus orígenes en el lenguaje de inteligencia artificial LISP, que se desarrolló a principios de los años sesenta para que los ordenadores pudieran tratar con más facilidad estructuras de datos complejas. Su nombre proviene del hecho de que es un lenguaje de "proceso de listas", lo que significa que su estructura de datos básica es una lista, en vez de una serie de caracteres o de una matriz numérica, como en el BASIC. Las funciones esenciales del LISP manipulan los datos dentro de una lista. Los elementos de la lista pueden ser símbolos simples o listas enteras. La ventaja de este enfoque es que de esta manera los datos no numéricos (como una oración) se procesan más fácilmente.

El LISP depende en gran medida del principio de la recursión, donde algo (por lo general una función o procedimiento) se define en términos de sí mismo. En el caso del LISP, el ítem que se define siempre es una lista. Éstas no son características accidentales del LISP, sino que surgen a raíz de sus orígenes en las investigaciones basadas en ordenadores sobre el lenguaje natural y la inteligencia artificial. Sin embargo, no se trata de un lenguaje fácil de aprender, y en 1968 un grupo de personas relacionadas con el Massachusetts Institute of Technology (MIT) se propuso desarrollar un lenguaje para niños basado en el LISP.

El líder del grupo del MIT era el carismático Seymour Papert. Anteriormente había estado estudiando durante algunos años el desarrollo cognosci-

Presentando el LOGO

El LOGO posee dos características fundamentales que hacen del mismo un lenguaje educativo tan eficaz. La primera es que es interactivo: cuando se digita una instrucción, inmediatamente se ven los resultados en la pantalla. Ello significa que es fácil ir avanzando (en especial para los niños y los principiantes) porque el alumno puede ir controlándose a sí mismo paso a paso.

La segunda característica esencial es que el LOGO es ampliable: operaciones completas se manipulan mediante listas de instrucciones de LOGO elementales. Estas listas se denominan *procedimientos*. Una vez que se ha definido un procedimiento como un conjunto de instrucciones determinadas, el nombre de ese procedimiento asume el estatus de una nueva instrucción de LOGO. A partir de entonces, el procedimiento entero se puede ejecutar simplemente digitando su nombre.

Al programar en LOGO muchas personas tienden a ser más exploradoras que en otros lenguajes. Algunas veces seguirán un enfoque bastante estricto y definirán un esquema específico desde el mismo comienzo. Otras veces empezarán con un problema central y escribirán un procedimiento para resolverlo, y luego construirán un programa alrededor de ese procedimiento. Se puede asumir un enfoque flexible al LOGO porque suelen existir varios caminos para llegar a un resultado determinado.



Creador y teórico
Seymour Papert, creador del LOGO, aparece en la fotografía durante un congreso que patrocinó Commodore en 1983. En la actualidad Papert está asociado con LOGO Computer Systems, Inc., (LCSI), que suministra programas de LOGO para el Sinclair Spectrum, los ordenadores Atari y otras máquinas



tivo (aprendizaje) en los niños pequeños con Jean Piaget (1896-1980), el eminente psicólogo educacional de su generación. Al pasar al MIT, Papert comenzó a trabajar en estrecha relación con un experto en inteligencia artificial, Marvin Minsky. En su trabajo con el LOGO Papert intentó unir las ideas de sus colegas, unificando teorías de aprendizaje cognoscitivo e inteligencia artificial.

El trabajo sobre el LOGO siguió durante los años setenta y se formaron otros grupos para experimentar con el nuevo lenguaje. El más notable de ellos fue el de Edimburgo. Todo su trabajo de desarrollo se llevó a cabo en departamentos de investigación universitarios utilizando ordenadores centrales o miniordenadores. El advenimiento de los micros supuso la difusión del LOGO a niveles más amplios.

El LOGO es un lenguaje sofisticado que requiere muchísima memoria, tanto para las instrucciones como para espacio de trabajo. Los intérpretes de LOGO de los micros típicamente exigen alrededor de 30 Kbytes de memoria, y otros ocho Kbytes o más para la visualización de gráficos. ¡Todo esto antes de que uno empiece siquiera a programar! En consecuencia, a pesar de que fue posible implementar intérpretes de BASIC simples en micros personales desde el mismo momento en que éstos se comercializaron, el LOGO para micros sólo se convirtió en una posibilidad factible cuando se pusieron al alcance del gran público los ordenadores personales de más de 48 Kbytes de RAM.

Pero fue *Mindstorms* (Basic Books, 1980), de Seymour Papert, la que sacó al LOGO de los departamentos de investigación y lo convirtió en foco de atención de un grupo mucho mayor de personas. En este libro, Papert desarrolla una visión de la forma en que se podrían utilizar los ordenadores en la educación. Este trabajo es el resultado de la síntesis de tres grupos de ideas: teorías del desarrollo cognoscitivo, inteligencia artificial y el movimiento en la educación orientado hacia el aprendizaje centrado en el niño. Lo que Papert desea es ver a los niños programando ordenadores, en vez de a los ordenadores programando al niño.

El libro postula la aparición de una nueva "cultura del ordenador", en la cual las ideas "formales" que previamente se consideraban fuera de las capacidades del niño podrían ser manipuladas por éste con facilidad. El niño sería capaz de hacerlo debido a la forma en la que habría utilizado los ordenadores para explorar ideas formales. Es esta exploración de ideas, activa, cooperativa y no estructurada (alumno-alumno y alumno-maestro), lo que constituye la "filosofía LOGO" sobre la que se basa la utilización del lenguaje en el campo de la educación.

Papert escribe y convence por el puro poder de su retórica. Sin embargo, en la teoría existen algunos problemas graves. Hay pocas pruebas experimentales que la sustenten, a pesar de algunos estudios realizados; las teorías de Piaget acerca del desarrollo cognoscitivo se convierten en una prescripción para la educación en forma que él jamás pretendió; y existen áreas de solución de problemas (¡incluso en matemáticas!) que el LOGO no cubre.

A medida que los maestros van utilizando más ampliamente el LOGO en la clase, están descubriendo que no todo funciona de la forma en que Papert lo describe y no están obteniendo los resultados que habían esperado. Existe el peligro del desencanto, pero una vez dejado de lado el entusiasmo excesivamente desproporcionado acerca de lo que el lenguaje puede hacer, el LOGO aún sigue siendo una excelente forma de introducir conceptos informáticos, de explorar ciertas clases de ideas y de desarrollar aptitudes para la resolución de problemas.

En los micros actuales el LOGO tiene muy poco espacio de trabajo y opera con excesiva lentitud. Hasta cierto punto es un lenguaje que está a la espera de que el hardware alcance sus niveles de exigencia. Pero como lenguaje para aprendizaje no tiene ningún competidor serio.



Versiones autorizadas
En la fotografía vemos las versiones más amplias y mejor documentadas del LOGO para el Commodore 64 (Terrapin-MIT), Sinclair Spectrum y ordenadores Atari (LCSI). Aunque caras, éstas son las versiones de LOGO que han sido autorizadas por los fabricantes y que se asemejarán en mayor grado al lenguaje MIT original. El LOGO para el Commodore 64 se vende en disco; el LOGO Atari viene en cartucho, y el LOGO Sinclair es una versión basada en cassette

¿Para quién es el LOGO?

¿A quién le puede ser útil aprender a programar en LOGO? Nosotros creemos que muchas personas, incluso los programadores experimentados, pueden aprender muchísimo a través de la programación en LOGO, incluyendo:

- Cualquiera que sea un recién iniciado en informática o en programación;
- Cualquiera a quien agrade jugar con los ordenadores y piense que su uso debe ser divertido;
- Quien se vea frustrado por una falta de poder expresivo en otro lenguaje de programación;
- Cualquiera que esté interesado en el pensamiento, el aprendizaje o la enseñanza;
- Quienes deseen una visión de áreas más avanzadas de la informática, especialmente quienes se dedican al estudio de la inteligencia artificial.

Habiendo dicho esto, debemos recordarle que, al igual que la programación en BASIC y en lenguaje máquina, el LOGO no es para todos. Específicamente, el LOGO podría no ser el mejor lenguaje para:

- Quien crea que utilizar ordenadores es "trabajar". Algunos lenguajes están diseñados para el trabajo, así como los caballos de tiro. Pero un caballo de tiro es lo último que uno escogería para una cabalgata por el campo;
- Quien desee o espere muchísima velocidad del ordenador para el proceso de sus instrucciones. El LOGO utiliza muchísima memoria y opera lentamente en la actual generación de micros. (En programas comparables, el LOGO puede funcionar a la mitad de velocidad que el BASIC.)

Incluso para estos grupos de personas puede ser valioso cierto conocimiento del LOGO: se puede usar para bosquejar una solución a un problema y prepararla para su traducción a otro lenguaje.

Pronto llegará...



la Tortuga

Uso de rutinas

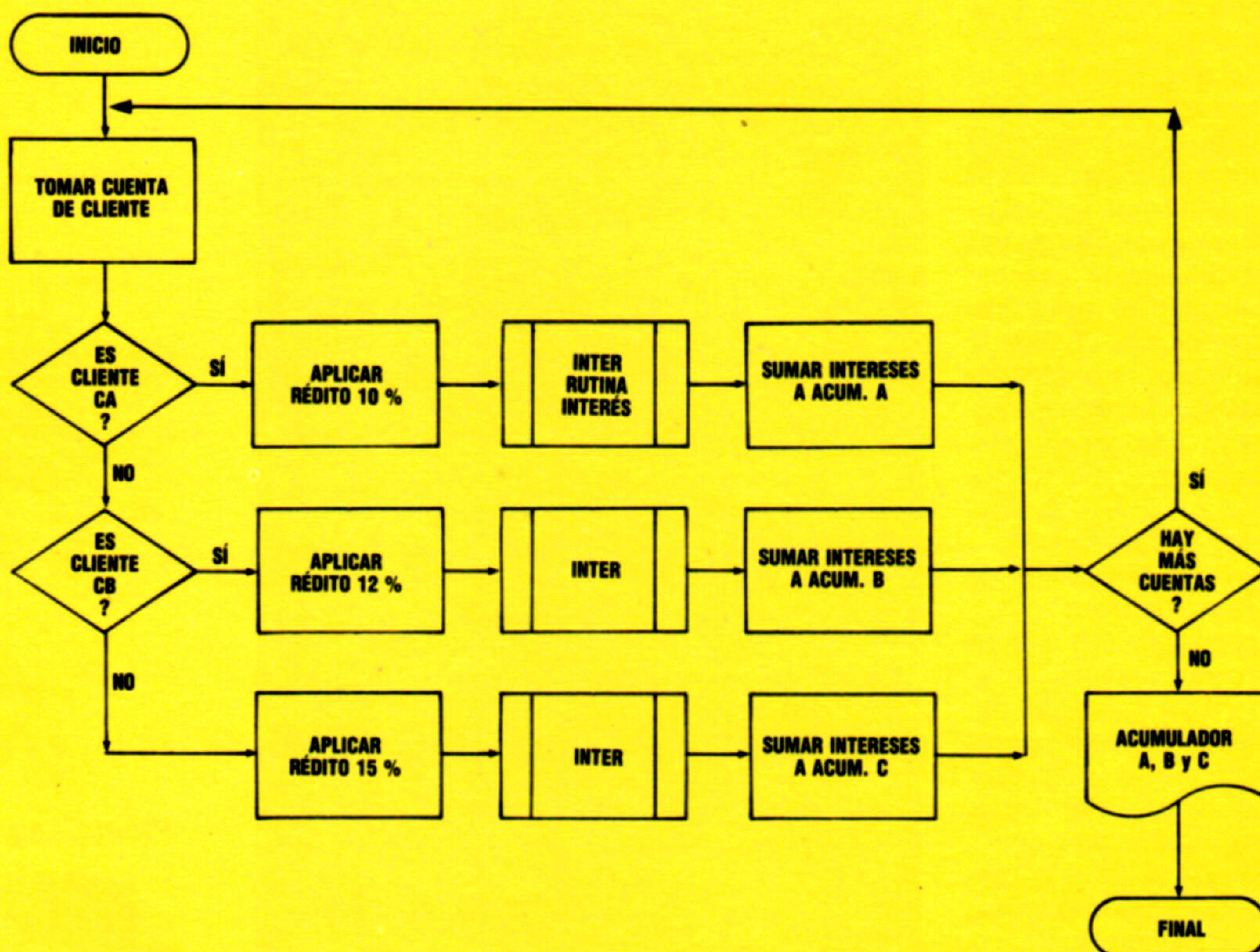
Ahora emplearemos una rutina en nuestro último ejemplo, para optimizar su solución



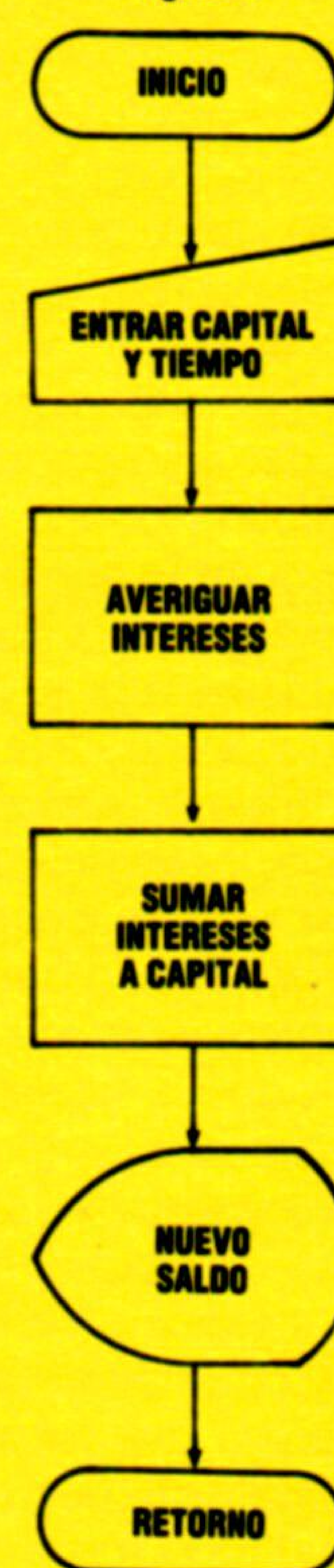
La solución dada al problema aparecido en el capítulo anterior mostraba cómo llegar al resultado deseado, pero a base de la repetición de una misma parte de la secuencia para cada uno de los tipos de cliente. Con anterioridad nos hemos referido a la utilidad de los bucles o bien a la agrupación en un solo punto de las partes repetitivas, pero en este caso en particular pudo comprobarse que tal alternativa no era viable (véase p. 968, fig. 2). A continuación se muestra cómo optimizar la solución mediante el empleo de una rutina.

Una rutina podría definirse como aquella parte del programa que, debido a que se repite en diferentes ocasiones a lo largo del mismo, se separa del cuerpo principal, indicándose su uso mediante un símbolo especial (véase arriba, a la izquierda).

Funciona de la siguiente forma. Supongamos que tomamos la cuenta de un cliente de la segunda categoría. Tras una respuesta negativa en la primera decisión, adoptará la vía correspondiente a cliente del tipo CB. Luego de darle valor al rédito (12), que es la única operación que no puede ser incluida en la rutina, el símbolo siguiente hace referencia al uso de la misma. Ésta recibe el nombre de INTER y está representada aparte. La rutina tiene su propio inicio, que es la conexión del programa principal con ella. También tiene su final, que en este caso no es físico, sino que retorna el control al programa justo en la instrucción siguiente a su llamada. Así, todas las veces que se pase por tal símbolo, el proceso continuará en la rutina, volviendo a su origen al final de ésta.



INTER





Nuevas expresiones

Los avances en la música electrónica son asombrosos: analizaremos aquí la codificación digital del sonido, o muestreo

A pesar de los muchos desarrollos que se han producido en este campo durante los últimos sesenta años, vale la pena recordar las características simples de voltaje controlado de un oscilador. Cuando un objeto físico cualquiera (ya sea el ala de una abeja, o una cuerda vocal humana) vibra, el aire circundante se expande y se contrae muy rápidamente, produciendo una forma de onda que el oído y el cerebro humano interpretan como sonido. Si a una diminuta lengüeta de metal se le aplica un voltaje eléctrico a través de un modulador (como una bobina de inducción de automóvil), el metal vibra, creando la forma de onda más simple: la onda sinusoide. La altura, o frecuencia de vibración, de la oscilación resultante depende del voltaje aplicado y, aunque en menor grado, de la densidad de la tira de metal. Esta diminuta unidad de generación de sonido se denomina *oscilador*. El control de voltaje ha sido el principal método para producir música sintetizada durante décadas.

La interface MIDI, que se anunció por primera vez en 1983, es una unidad que está diseñada para permitir que un sistema digital (como un ordenador) controle a otro, como, por ejemplo, un sintetizador. Su desarrollo es consecuencia de los adelantos que se vienen sucediendo en la producción de música electrónica en la última década.

Durante varios años los estudios de grabación han incorporado muchas piezas diferentes de equipo de proceso de sonido; con frecuencia se considera como prueba concluyente de la valía de un estudio una impresionante gama de unidades de filtro y reverberación. Del mismo modo, en las actuaciones en vivo de un músico de los años setenta que se sirviera de un sintetizador, éste debía estar enteramente rodeado por bancos de teclados, cada uno de ellos con multitud de controles.

Al considerar lo que sucede en un estudio de grabación bien equipado, es útil pensar en el juego de sociedad "Susurros chinos". En este juego se va pasando una frase de persona a persona. El último jugador recita luego lo que él ha escuchado de la oración original. Así, una frase sencilla y directa puede haberse convertido en un conjunto de palabras sin sentido o con éste totalmente distorsionado. El proceso que tiene lugar en el estudio de grabación es similar, pero en este caso la frase original es un conjunto de sonidos musicales. Y en lugar de una cadena de oyentes, cada uno de los cuales transmite una versión tergiversada del original, aquí hay un grupo de unidades para proceso de sonido, cada una de ellas controlable y con una tarea específica a cumplir. Cualquiera que se sirva de este equipo probablemente utilizará un tablero de conexiones centrales para hacer las conexiones o, de lo contrario, conectará las unidades entre sí directamente. Un ingeniero de sonido experimentado puede calibrar los controles de cada unidad de forma manual en cuestión de segundos, pero es fácil imaginar los problemas de sincronización y comunicación inherentes a esta clase de conexiones.



Trío innovador
Alannah Currey, Tom Bailey y Joe Leeway (de izquierda a derecha), de los Thompson Twins. En sus comienzos un grupo *arty* de siete componentes, actualmente actúan como trío con el apoyo de ritmos secuenciados grabados en cinta. (Se suele denominar *arty* a aquel conjunto musical cuyos integrantes en su mayoría provienen de escuelas de arte o conservatorios.)

El músico "de teclado" de los años setenta tenía un problema diferente. En su caso, la dificultad inmediata no era cómo producir un sonido con cada instrumento en sucesión: el músico sólo tenía que mover la mano de un teclado a otro. Lo más probable era que se le presentaran dificultades cuando los dos teclados se tenían que tocar al mismo tiempo, pero los músicos de clavicordios y de los órganos de las iglesias supieron cómo hacerlo durante siglos. Incluso música tan compleja como una fuga de Bach, se podía ejecutar en un solo teclado y, de todos modos, la mayor cantidad del "trabajo" de estudio no se hacía en una sola "sesión". En cambio, cada parte del sintetizador se grababa sola, superponiendo las partes subsiguientes sobre la primera, utilizando distintos sectores de una cinta de canales múltiples. Pero ejecutar este tipo de música requería un artista diestro, o dos músicos corrientes, o bucles de cinta y un ingeniero de sonido.

Pero el desarrollo más importante de los años setenta fue el de presentar las unidades generadoras de sonido *dentro* del sintetizador, y las técni-



Centro de control
La interface Roland MP401 MIDI es una sofisticada unidad que conecta sintetizadores digitales a microordenadores. Controla todas las funciones de sincronización externa, el tiempo interno y externo, la entrada/salida de cinta y la salida del sintetizador. Ello significa que el ordenador anfitrión sólo es responsable del envío y la recepción de instrucciones y de la gestión de la memoria. La MP401 funciona con el IBM PC y el Apple IIe. Según Roland, en un futuro cercano habrá una versión disponible para el Commodore 64

cas instrumentales en general. Un buen ejemplo de esto es el nacimiento del sintetizador de bajos controlado por teclado. En los años sesenta, el estilo Motown de música pop-soul dependía en gran medida de los bajos eléctricos. Durante los años setenta los bajistas *funk* alcanzaron un nivel de virtuosismo que rivalizaba con el de los principales guitarristas; a finales de la década este estilo ya no pudo evolucionar más y surgió el sintetizador de bajos controlado por teclado. Esto supuso nuevos problemas al músico de teclado, a quien ahora se le exigía que asumiera las funciones del bajista (trabajando con el tambor para "sujetar" la sección de ritmo en un tiempo estricto) y que ahora no estaba tocando música de "teclado". A medida que aparecieron en el mercado nuevos sintetizadores analógicos se dispuso de sonidos de trompeta, saxofón y tambor. Más y más músicos de teclado se volcaron a un dispositivo simple para asumir todas estas nuevas responsabilidades. Éste era el *secuenciador*.

Un secuenciador es un dispositivo que coordina varios ramales de música independientes unificándolos en una pieza única según un patrón especificado. Opera mediante la utilización de voltajes controlados que se generan a partir de un oscilador para producir una serie de tonos de frecuencias diferentes. Cuanto mayor es el voltaje, más rápido vibra la tira metálica y la forma de onda resultante se escucha como un sonido agudo. Para controlar el oscilador se utiliza una unidad de secuenciación. Ello es necesario porque la música muy raramente se compone de frases ininterrumpidas; se requieren breves vacíos o largos silencios, tanto para crear patrones rítmicos como para conformar la estructura global de la música. Un vacío en un patrón de frases se produce cuando la unidad de control le envía al oscilador un voltaje cero. La secuenciación tiene por objeto asegurar que el vacío se produzca en el mismo lugar cada vez que se repite el patrón.

A fines de los años setenta eran pocos los sintetizadores que tenían facilidades completas de secuenciación, pero los músicos fueron rápidos para utilizar aquello de lo que disponían. La vibrante música disco que produce Giorgio Moroder con Donna Summer es en gran parte música de secuenciador, y los nuevos grupos británicos de sintetizador desarrollaron un estilo completamente nuevo para ponerlo a la altura de los nuevos equipos. Ya no es necesario tocar cada nota individual con una serie de gestos grandilocuentes, al estilo de Rick Wakeman. En cambio, toda una secuencia, o *riff* (término que podríamos definir como "conjunto de notas definitorias o características de una canción"), se podía iniciar o interrumpir modificando un control. En el ínterin el músico se podía alejar

Steve Cross

Contar y tocar

Se pueden producir sonidos electrónicos programando la salida de tono y/o generadores de ruido blanco (así es como uno crea sonido en un microordenador), o también mediante el muestreo de sonidos reales, haciendo un "modelo" digital de los sonidos y utilizando ese modelo para recrear la forma de onda de los sonidos a través de un convertidor de digital a analógico que active un altavoz. El muestreo proporciona una imagen de sonido exacta (según la cantidad de muestras de forma de onda almacenadas), que de lo contrario sería difícil de programar excepto mediante ensayo y error: ¡intente sintetizar en su micro el ritmo de un tambor pequeño!



Analógico
La salida del micrófono es simplemente una forma de onda de tiempo de voltaje: una analogía eléctrica del sonido. La aplicación de esta forma de onda a un altavoz reproduce el sonido

Digital
El convertidor de analógico a digital "digitaliza" la forma de onda continua en una secuencia de mediciones de voltaje discretas. Cuantas más mediciones de éstas se tomen, más exacta será la imagen digital del sonido







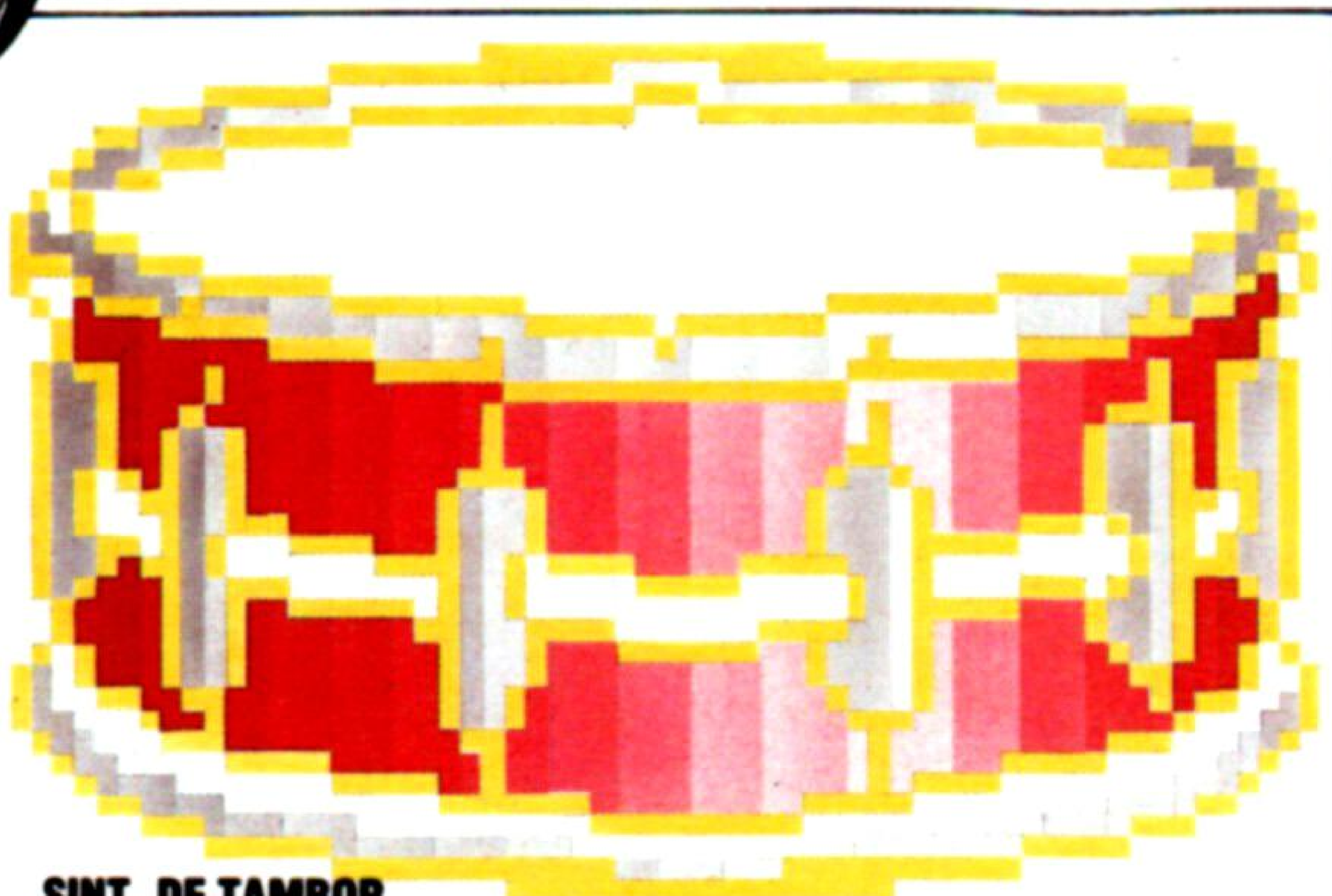
del teclado, bailar al ritmo del compás del secuenciador y después regresar a otro teclado para tocar una melodía o una serie de acordes. A mediados de los ochenta, todo el estilo de actuación de un grupo como los Thompson Twins está determinado por la existencia del secuenciador.

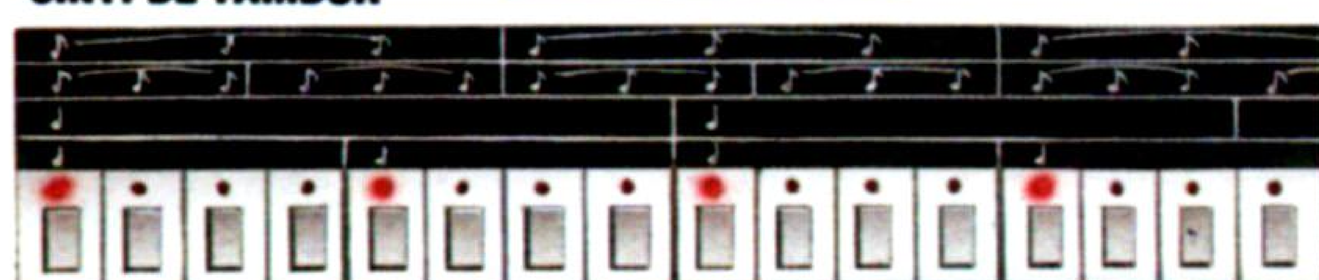
Cuando aparecieron por primera vez los sintetizadores digitales, a menudo su diseño se modelaba en función de sus predecesores analógicos. Los músicos descubrieron que los nuevos instrumentos desarrollaban aún más sus "aptitudes de secuenciación", y es en esta área del control digital donde se ha generado el mayor interés. Ello lo demuestra la reciente popularidad de la batería eléctrica Linn, una de las primeras unidades en utilizar el sonido muestreado, en este caso proporcionado por Steve Gadd, el mejor batería norteamericano. En la máquina Linn, los patrones de tambor se graban en forma digital del mismo modo en que los datos del ordenador se graban en discos flexibles. Las secuencias resultantes de unos y ceros se codifican entonces en chips de ROM. Mediante el acceso a un chip determinado, un músico o un productor puede reproducir el sonido original como si se lo estuviera tocando en vivo. La gran ventaja de digitalizar el sonido es que la salida puede ser alterada en la consola, apartándose del patrón original en cuanto a tiempo, ritmo, volumen, etc.

Ahora nuestros dos ejemplos originales (el estudio de grabación y el músico con sintetizador en el escenario) tienen ciertas características en común. Además de grabar música, el trabajo en el estudio refleja el *boom* del video que se ha producido en estos últimos años. El video ha aportado un nuevo estilo y una nueva conciencia en la fabricación de imágenes: los productores exigen que la música que lo complementa refleje este hecho.

Si la música que acompaña a un video se produce mediante instrumentos convencionales, la sincronización con la imagen en pantalla es muy similar a la que se realiza con una película. Sin embargo, si la música se compone de sonidos individuales y frases producidas por sintetizadores digitales, hay muchas cosas que pueden ir mal. Imaginemos que en un determinado video una vasija con flores se cae al suelo, donde se hace añicos. El músico que trabaja en la partitura ha creado una frase que se acelera hasta el punto en el que la vasija cae, y ha dispuesto un acorde de percusión para el momento en que se estrella contra el suelo. Comienza la grabación y empieza la frase, pero en seguida resulta evidente que la velocidad de la aceleración se ha calculado mal y la frase concluye mientras la vasija todavía está sobre la mesa. El músico prueba entonces el acorde de percusión, que se graba en un canal distinto de la cinta. La grabación se reanuda, y en esta ocasión el acorde se graba una fracción de segundo más tarde. Los músicos necesitan una forma de conjuntar los instrumentos digitales de modo que todo suceda en el momento adecuado. Lo que se precisa, entonces, es una interface digital.

El músico de sintetizador tiene problemas similare, esta vez en una actuación en directo. Su equipo incluye dos sintetizadores digitales, producidos por diferentes fabricantes, y una batería eléctrica o caja de ritmos Linn. El músico tiene preparadas frases en un sintetizador y en el Linn, pero como éstos no llevan juntos el compás, suele acabar por dejar que el Linn funcione automáticamente mientras toca el otro de forma manual. El resultado es que su segundo sintetizador, adquirido en razón de la calidad de sus sonidos previamente preparados, queda sin intervenir. Este músico necesita una forma de unir sus instrumentos de modo que el secuenciado de todo el material se produzca en el punto correcto. Además es indispensable que el secuenciador de su primer sintetizador toque los sonidos preestablecidos del segundo. Por otra parte, sea cual fuere el equipo que utilice, debería ser aplicable a algo más que a sus propios sintetizadores: ¡bien podría nuestro músico descubrir su verdadera identidad en el estudio a que hemos aludido anteriormente, que planteaba tantos problemas en la sincronización de video!

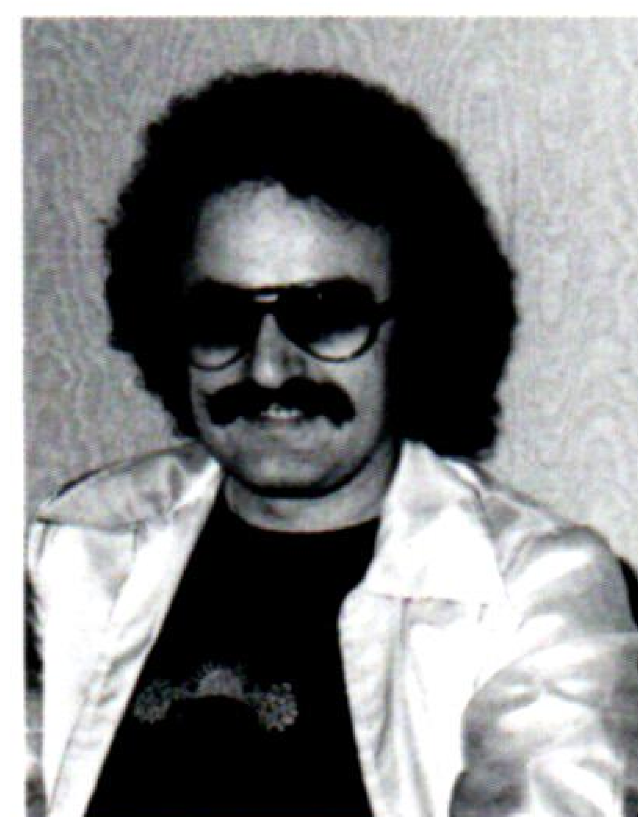




Ritmo electrónico
Se puede construir una frase de tambor en cualquier orden en las teclas del sintetizador. Al reproducirla, un explorador (*scan*) barre el teclado a una clave de tiempos preestablecida, tocando un compás de tambor digitalizado si esa tecla ha sido pulsada e iluminando su LED. La frase se puede entonces crear y editar estrato por estrato







En vanguardia
Donna Summer, con el productor Giorgio Moroder, fue una de las primeras artistas pop que utilizó en la música grabada sintetizadores y ritmos producidos electrónicamente

El factor humano

En el diseño de programas es esencial la interface hombre-máquina, que se ocupa del intercambio de información entre usuario y programa

Durante muchos años la programación de ordenadores ha sido un tema misterioso comprendido sólo por profesionales que podían dedicar mucho tiempo y esfuerzo a la materia. Antes del advenimiento del microordenador con su teclado tipo máquina de escribir, con frecuencia los programas eran entrados byte a byte a través de interruptores situados en el panel frontal del ordenador, o perforando agujeros en cintas en la consola de un teletipo.

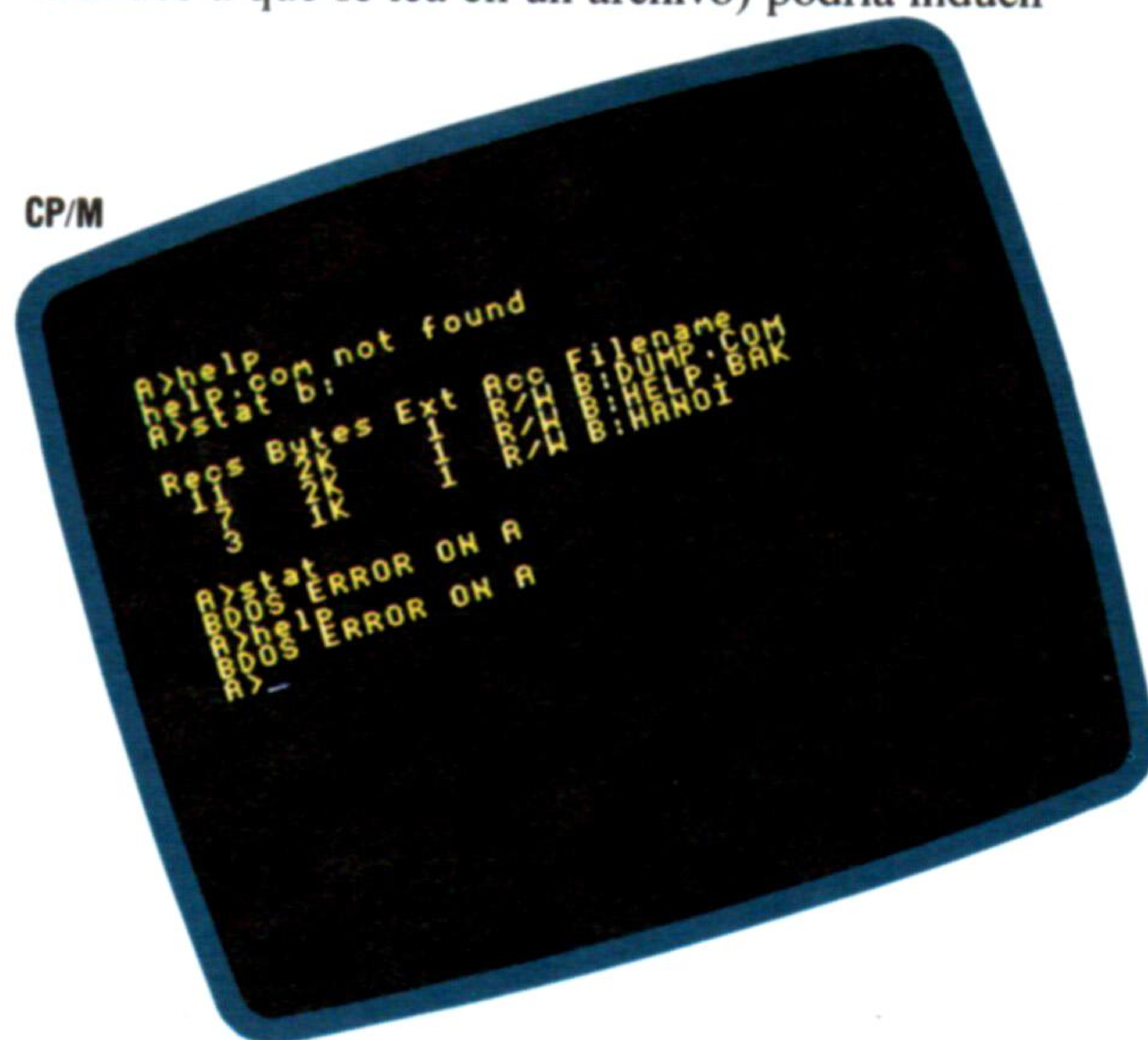
En comparación, el usuario es actualmente una criatura mimada. Los fabricantes ya no esperan que el propietario de un ordenador luche con el lenguaje máquina, y se ha acuñado la frase "amabilidad hacia el usuario" para indicar que cualquiera, independientemente de su experiencia, puede utilizar y programar micros. En 1982 el Alvey Committee, en el informe *A programme for advanced information technology* (Programa para la tecnología avanzada de la información), identificaba la interface hombre-máquina (MMI: *man-machine interface*) como una de las cuatro áreas principales de investigación y desarrollo, junto con la ingeniería de software, el diseño de circuitos integrados a muy gran escala (VLSI: *very large scale integration*) y los sistemas basados en el conocimiento.

En cualquier aplicación la interacción entre el ordenador y el usuario, por la cual se pasan entre uno y otro datos o instrucciones, reviste una importancia capital. Este "diálogo" se conduce a través de los dispositivos de entrada/salida (E/S) del ordenador, con el teclado actuando como principal fuente de entrada y la pantalla de visualización proporcionando la salida. Palancas de mando, lápices ópticos, "ratones", pantallas al tacto y otros dispositivos también se pueden utilizar para la entrada, mientras que el ordenador puede servirse de una impresora, un generador de sonido (o de voz) o incluso un robot para manifestar la salida.

Además de cualquiera de las limitaciones que pueden plantear los dispositivos de E/S utilizados, el diálogo entre usuario y máquina se ve influido por el software. Por ejemplo, el sistema operativo (OS) del ordenador controla muchos detalles de la operación del teclado y la pantalla. La velocidad a la cual se repiten las teclas cuando se mantienen pulsadas y la demora entre repeticiones las establece el sistema operativo, que también se guarda las pulsaciones de teclas para permitir al ordenador almacenar caracteres que se han digitado más rápidamente de lo que los mismos se pueden visualizar. Esto es importante porque tiene relación directa con la velocidad a la cual el usuario puede introducir información. El tamaño del buffer es vital y el usuario debería conocerlo; el sistema operativo CP/M, por ejemplo, guarda en el buffer una única pulsación de tecla, mientras que muchas máquinas personales almacenan 10 pulsaciones o más.

Pero los buffers de tecla pueden plantear problemas. Un usuario experimentado que esté trabajando con un sistema activado por menú puede saber de antemano que las opciones de menú que él requiere son la 2 del menú principal, la 5 del menú siguiente, luego la 3, 4, 6, etc. Dado que él está familiarizado con el sistema, entra sus opciones a gran velocidad. Con un buffer de 10 caracteres, el usuario terminará allí donde quiera, porque todas las pulsaciones de teclas se "recordarán" en la secuencia correcta. Con un buffer de un solo carácter, el tiempo necesario para visualizar el segundo menú puede que sea mayor que el tiempo que se precise para digitar la secuencia entera de opciones. Por consiguiente, en vez de seleccionar la opción número 5 de este menú, después la 3 y así sucesivamente, sólo se realiza la opción número 6 (porque éste es el único carácter retenido en el buffer) y el sistema se detiene allí.

Pero un buffer grande también puede suponer algunos problemas. Un programa de menú que demore mucho tiempo para reaccionar ante una pulsación de tecla (lo que puede ocurrir si la opción conduce a que se lea en un archivo) podría inducir al usuario a pensar que no está pasando nada. La respuesta natural en este caso sería probar la última opción introducida, pulsando luego una selección de teclas hasta que hubiera una respuesta. Esto podría llevar a que el programa intentara procesar los caracteres ilegítimos retenidos en el buffer; ¡los resultados podrían ser sorprendentes!



CP/M

La "recogida de basura", que implica limpiar los registros de memoria del ordenador para dejar espacio de trabajo libre, constituye otra fuente de problemas. Esta operación puede hacer que un programa parezca "colgado" durante largos perío-

dos, en los cuales el usuario puede nuevamente tratar de emprender una acción correctiva. Es probable que la recogida de basura produzca problemas en programas largos que realicen mucha manipulación de series. Algunas versiones de BASIC permiten que el programador provoque una recogida de basura; es una buena idea hacerlo a intervalos frecuentes, pero es conveniente transmitir al usuario un mensaje "espere por favor", dado que parecerá que el ordenador no esté haciendo nada mientras se ocupa de la recogida de la basura.

La forma en que un lenguaje de programación manipule la entrada y la salida influirá en el diseño de la interface entre el ordenador y el usuario. Las superiores facilidades para manipulación de series del BASIC conducirán a un ejemplo más sofisticado de series en el diálogo hombre-ordenador que el que permiten lenguajes como el PASCAL. Los BASIC que posean instrucciones incorporadas para posicionamiento del cursor favorecerán trazados de pantalla mejores que aquellos que no las posean. Lo mismo puede decirse de los BASIC con instrucciones para gráficos. El BASIC está bien provisto de instrucciones de entrada/salida; INPUT y PRINT están bien para programas sencillos. Pero para un verdadero control de entrada, intente experimentar con

"Log-on" (Conexión) al Micronet 800



GET$, INKEY$, INPUT$() e instrucciones similares. PRINT USING es una instrucción sumamente versátil para formatear la salida; tiene un incalculable valor para alinear puntos decimales y para justificar columnas de texto.

En cualquier sistema hombre-máquina, el elemento más impredecible es el usuario. Sin embargo, a diferencia de cualquier otro componente, el usuario posee ciertas características de rendimiento que se han de comprender antes de diseñar la interface.

Las personas comparten con los ordenadores la característica básica de ser "procesadores de información". No obstante, los seres humanos tienen limitaciones inherentes respecto a la cantidad de información nueva que pueden retener en la "memoria de trabajo"; se ha comprobado que para la mayoría de ítems de información se pueden retener simultáneamente en el cerebro alrededor de siete ítems diferentes. El tamaño de estos ítems depende de lo significativos que sean o de lo bien estructurados que estén. Si la información a recordar se compone de caracteres al azar, cada ítem se compondrá de no más de un único carácter. Pero si los caracteres no son al azar sino que son apellidos comunes, cada ítem recordado podría ser un nombre completo. Al incrementar la estructura de la información de esta manera, se incrementa la capacidad del usuario para recordarla y utilizarla.

Existen varias formas de ayudar a la gente a estructurar la información cuando utilizan ordenadores. Un método consiste en relacionar los datos con estructuras familiares y bien comprendidas; ésta es la forma en la que funciona la visualización del "escritorio" estilo Lisa. Del mismo modo, se podría organizar un paquete de hoja electrónica financiera como para que tuviera el aspecto de un libro, con páginas, índices, etc. Otro procedimiento consiste en entrenar al usuario para que comprenda estructuras que no le sean familiares. Mostrando ejemplos repetidamente y explicando temas en profundidad, se podría utilizar el propio programa para enseñarle al usuario cómo se debería estructurar la información. El entrenamiento de esta clase tiene el inconveniente de que resulta caro, tanto en tiempo como en esfuerzo. Instrucciones detalladas, pantallas de "ayuda" y "postes indicadores" pueden proporcionar un tipo de entrenamiento en línea, pero son difíciles de utilizar con eficacia.

Apple Macintosh



Por último, presentar la información en patrones reconocibles puede ayudar al usuario a comprender el programa. Esto se puede hacer valiéndose del color o del trazado para guiar el ojo hacia la información deseada. Para entender lo que esto significa, consideremos la codificación con colores como la que utilizan el Prestel y programas de videotexto similares. En una página típica, el encabezamiento y el pie estarán dispuestos en bloques del mismo color; habrá un único color de fondo y el texto se visualizará en otros dos colores, alternando un párrafo en cada color. Las palabras clave se pueden realzar aplicando aún otro color. Todo ello tiene por finalidad permitir que el usuario seleccione sólo la información requerida y que ignore secciones enteras de la página si éstas contienen información que carece de un valor inmediato. Sin embargo, la codificación con colores puede ser fuente de confusión si se la emplea con exceso.

Tres niveles
Estas fotografías de sistemas operativos de microordenadores ilustran tres niveles variables de amabilidad hacia el usuario. En la primera fotografía (de izquierda a derecha) un usuario nuevo está intentando comunicarse con el sistema operativo CP/M. El CP/M no posee facilidades de "ayuda" incorporadas, de modo que para que se lo pueda utilizar adecuadamente es necesario un profundo conocimiento de las instrucciones. Nuestro segundo ejemplo es un sistema conducido por menú: el menú "Log-on" para el Micronet 800 en el BBC Micro. Las opciones están claramente numeradas y el usuario realiza su elección entrando el número apropiado que se le indica en el menú. La pantalla no ofrece gran cantidad de información, de modo que el usuario debe comprender las opciones para poder hacer uso de ellas. Nuestra tercera fotografía muestra el sistema operativo del Apple Macintosh, que proporciona indicaciones visuales y visualizaciones gráficas, así como menús sencillos y fácilmente comprensibles

Abierta al mundo

Cómo acceder a la puerta para el usuario con el fin de controlar fenómenos físicos externos al ordenador

Muchos de los micros personales poseen puertas para el usuario que permiten la entrada al mapa de memoria del ordenador en virtud de una serie de conexiones eléctricas. La base de todos los sistemas digitales es que el sistema binario de unos y ceros se pueda representar mediante dos niveles de voltaje. Normalmente, un cero se representa mediante 0 voltios y un uno mediante +5 voltios.

Cada posición de memoria se compone de un grupo de ocho celdas individuales, teniendo cada una de ellas un nivel de voltaje de 0 o 5 voltios. El patrón de estos niveles de voltaje determina, por consiguiente, el número que está almacenado en esa posición de memoria. Si una celda posee un nivel de voltaje de +5 voltios, decimos que está *activada (set high)*, y si la celda posee un nivel de 0 voltios, se dice que está *desactivada (set low)*. Las conexiones externas de la puerta para el usuario están unidas eléctricamente a una o más posiciones de la memoria del micro y mediante la lectura de los valores de estas posiciones, o mediante la escritura de valores en ellas, podemos monitorizar o controlar sistemas eléctricos fuera del ordenador.

Existen dos tipos de conexiones para puerta para el usuario. Algunas puertas tienen grupos de patillas separados (ocho para entrada y ocho para salida) conectadas con dos posiciones de memoria. Otros utilizan las mismas ocho patillas tanto para entrada como para salida. Aquí vamos a analizar el segundo tipo de configuración, que es el que emplean los micros BBC y Commodore 64.

Registros de dirección de datos

Además de tener una posición conectada con las ocho patillas de la puerta para el usuario, los micros con puertas bidireccionales hacen uso de una segunda posición de memoria, conocida como el *registro de dirección de datos* (DDR: *data direction register*). Este registro determina si cada una de las ocho líneas ha de enviar o recibir datos. Un uno en el DDR coloca a una línea en modalidad de salida, y un cero la coloca en modalidad de entrada. Para situar las ocho líneas de la puerta para el usuario en

Enchufando

Comenzamos nuestro proyecto haciendo cables para el BBC Micro y el Commodore 64. Estos cables se utilizarán para conectar las máquinas con el exterior a través de un formato común de ocho líneas de datos con una línea a tierra a cada lado. Se necesitará:

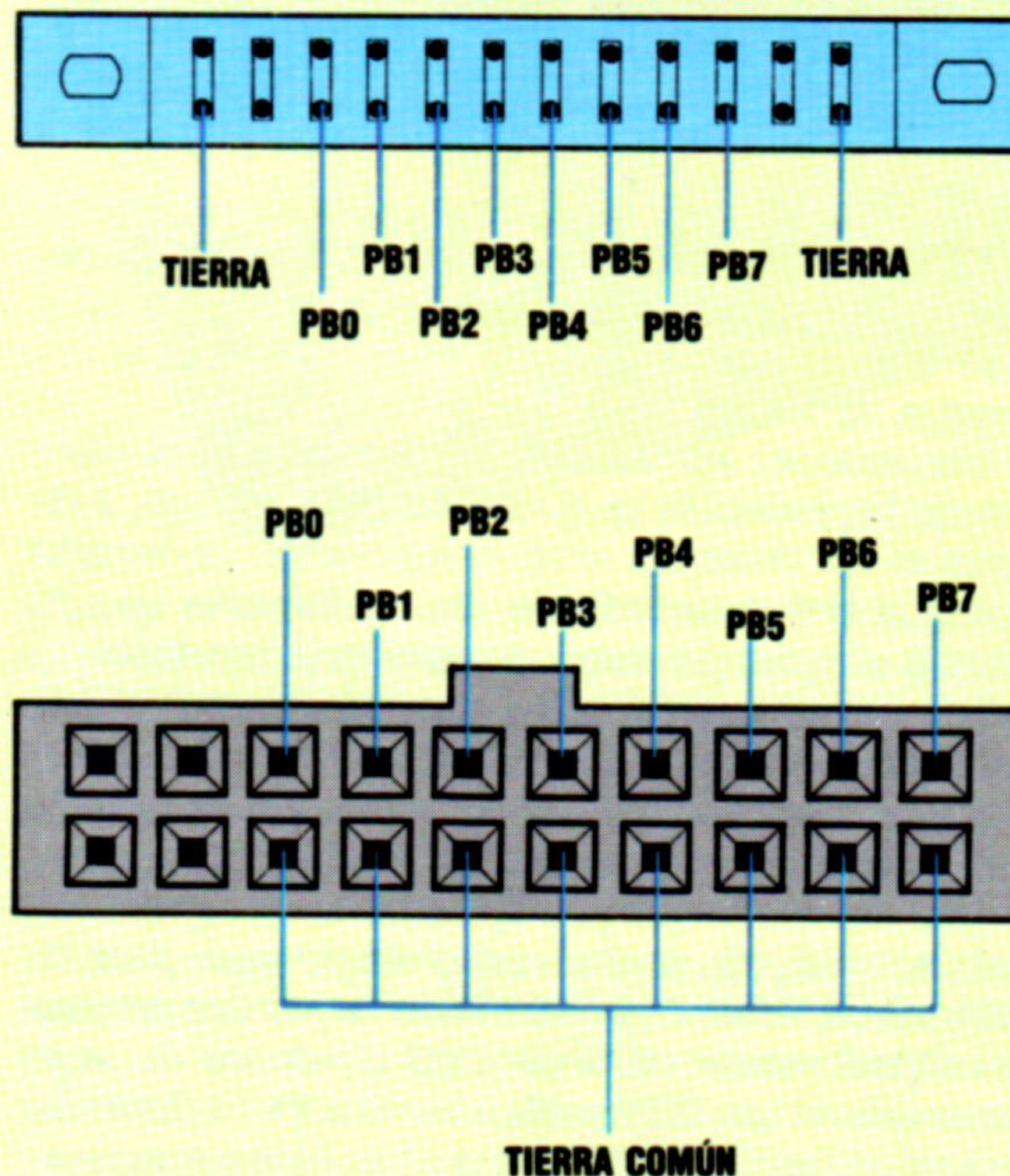
BBC Micro
- enchufe IDC de 20 vías
- cable plano IDC de 20 vías (un metro)
- estaño para soldar y soldador

Commodore 64
- conector marginal de 0,15" y 24 vías
- cable plano de 10 vías (alrededor de un metro)
- estaño para soldar y soldador

Los diagramas de las patillas de la puerta que vienen en los manuales de las máquinas muestran la posición de las 10 líneas (dos a tierra, ocho de datos) que necesitamos. El enchufe IDC del BBC posee una patilla de posicionamiento a uno de los lados que se separa en dos partes desiguales: sostenga el enchufe verticalmente con la patilla lejos de usted y su sección de ajuste arriba, y pase por la misma más o menos una pulgada de cable plano con la franja roja a su derecha. Cierre el enchufe sobre el cable apretándolo firmemente (tal vez con una prensa de tornillo o tornillo de ajuste). Separe y recorte 10 líneas por el otro extremo según la ilustración, luego pele y estañe los extremos de las líneas restantes (véase p. 524).

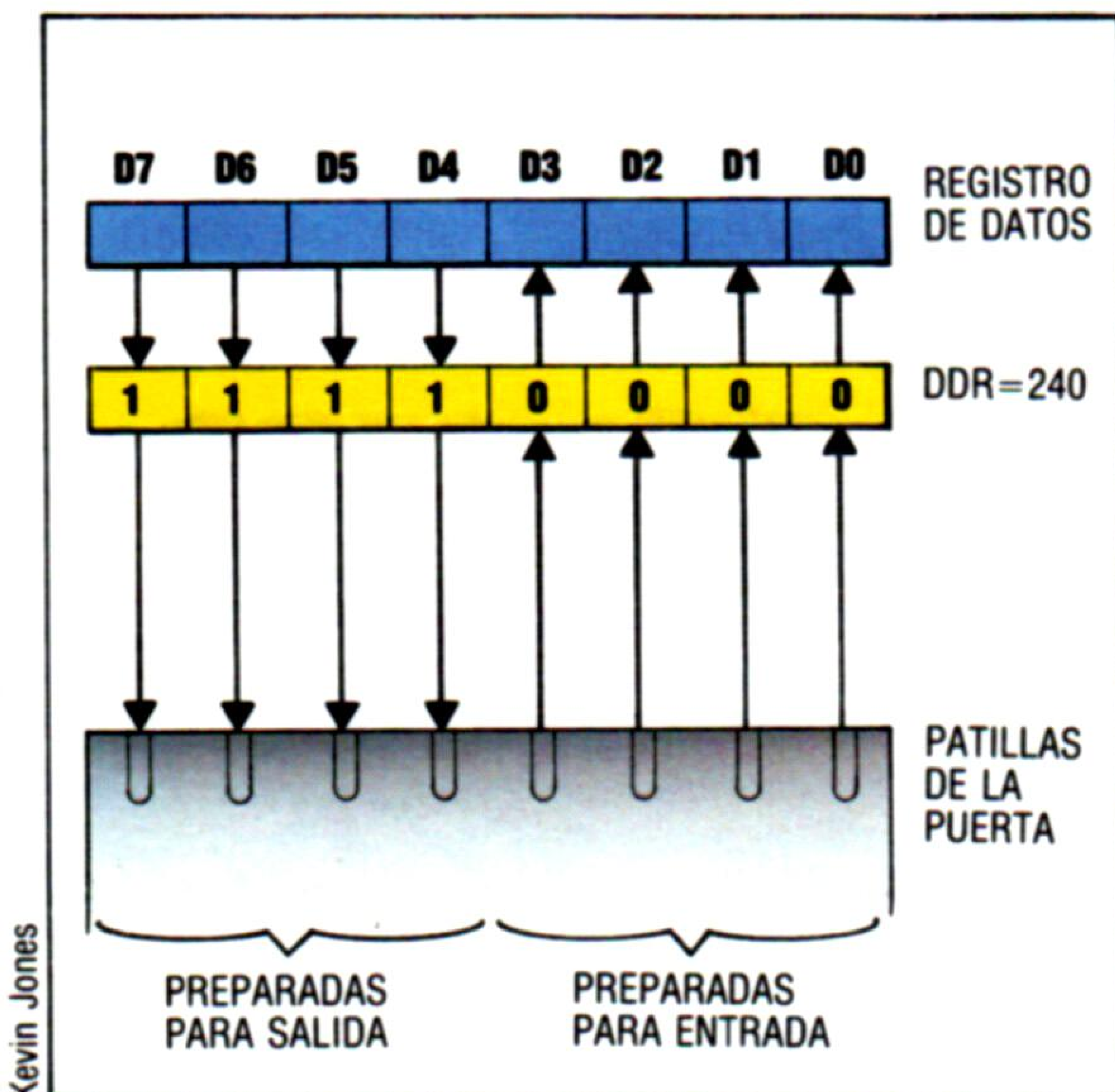
Para el Commodore 64, marque uno de los lados del enchufe claramente como la parte superior (y al conectarlo a la máquina tenga siempre este lado hacia arriba). Pele y estañe los dos extremos de las 10 líneas y suelde el cable a las patillas inferiores del conector marginal.

Utilice los programas de prueba y un tester (véase p. 567) para comprobar sus cables





salida, el DDR habrá de establecerse en 255 (es decir, 11111111 en binario). Del mismo modo, las ocho líneas se pueden establecer para aceptar entrada estableciendo el valor del DDR en cero. Las ocho líneas se pueden configurar en cualquier combinación de líneas de entrada o salida estableciendo en el DDR el valor apropiado. Por ejemplo, las cuatro líneas más significativas de la puerta para el usuario se podrían establecer en modalidad de salida, y las cuatro menos significativas en entrada, colocando en el DDR el valor 240 (es decir, 11110000 en binario).



Los registros de datos y de dirección de datos poseen las siguientes direcciones:

Tipo de micro	Registro de datos	Reg. de direc. de datos
BBC Micro	&FE60 (65120 dec.)	&F362 (65122 dec.)
Commodore 64	$DD01 (56577 dec.)	&DD03 (56579 dec.)

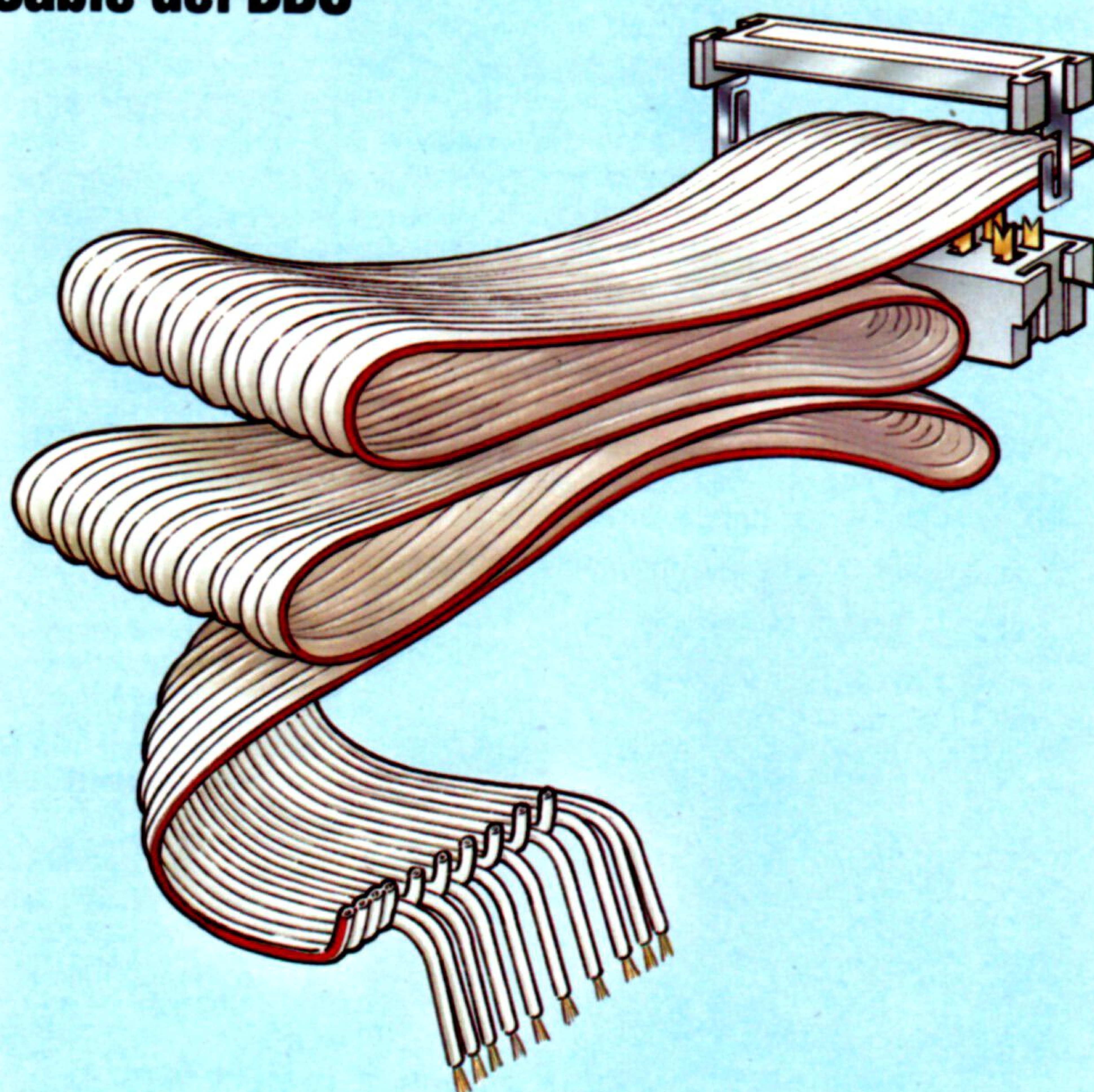
El siguiente programa coloca la puerta para el usuario de manera que se puedan usar las ocho líneas para entrada, y visualiza el registro de datos:

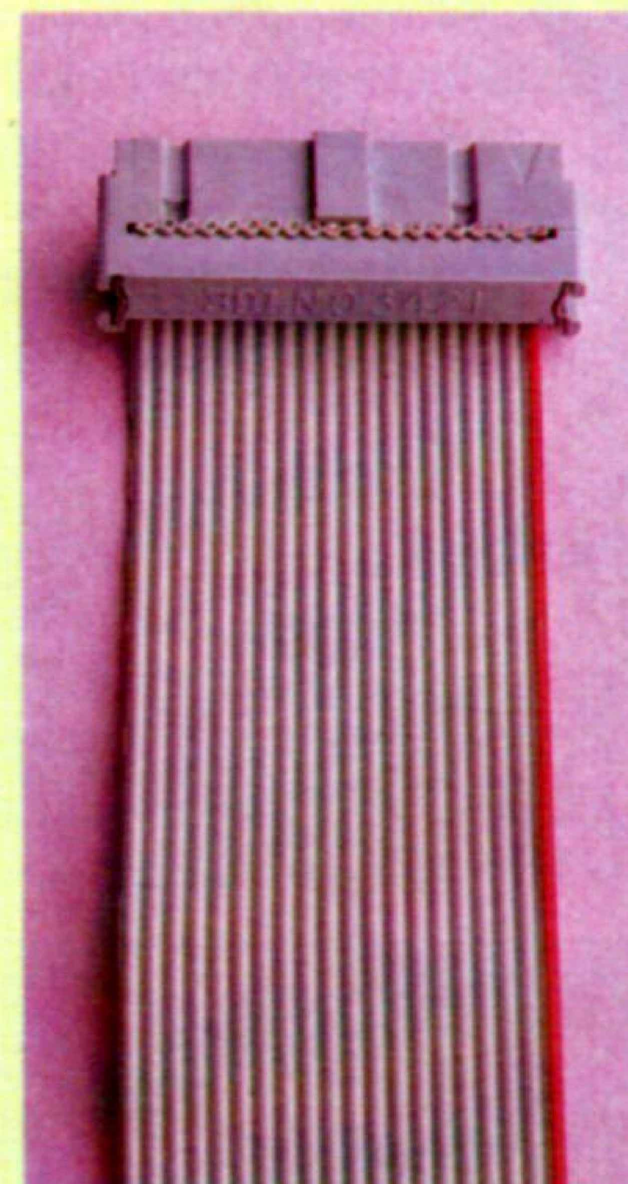
```
4 REM ********** C64 **********
5 REM *  VISUALIZACION REGDAT  *
6 REM ********** C64 **********
10 DIM A$(10):A$(0)="E":A$(1)="H"
20 REGDAT=56577:DDR=56579
30 POKE DDR,0:REM=ENTRADA SOLO
50 :
100 PE=PEEK(REGDAT):GOSUB 500
150 PRINT "REGDAT=";PE;"=";B$
200 GOTO 100
300 :
499 REM ****************************
500 REM *   S/R CONVERSION BINARIO  *
501 REM ****************************
550 B$="":N=PE
600 FOR D=1 TO 8
650 N1=INT(N/2):R=N-2*N1
700 B$=A$(R) + B$:N=N1
750 NEXT D:RETURN
```

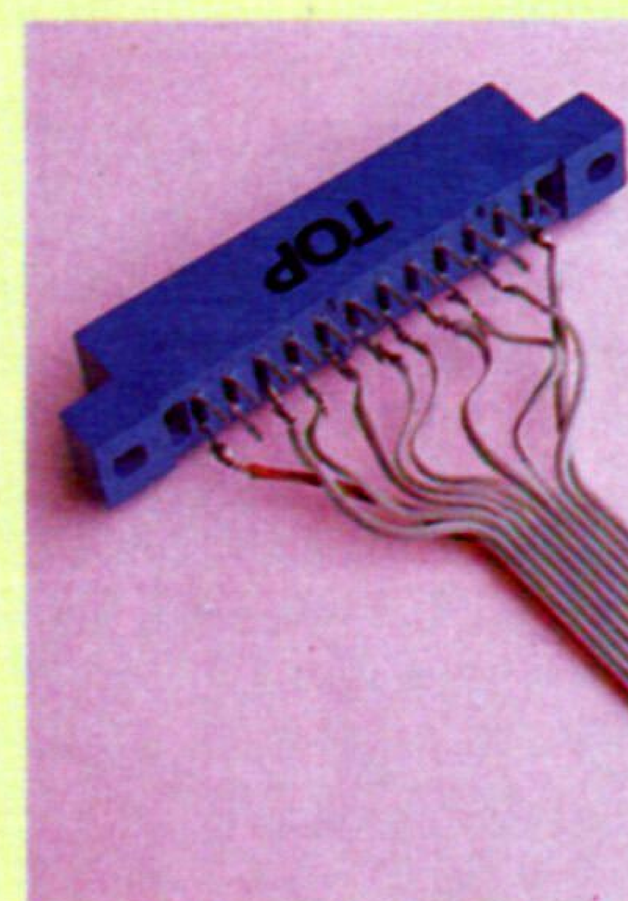
Para el BBC, introduzca estas modificaciones:

```
20 REGDAT=&FE60:DDR=&FE62
30 ?DDR=0
100 PE=?(PE):GOSUB 500
```

Cable del BBC



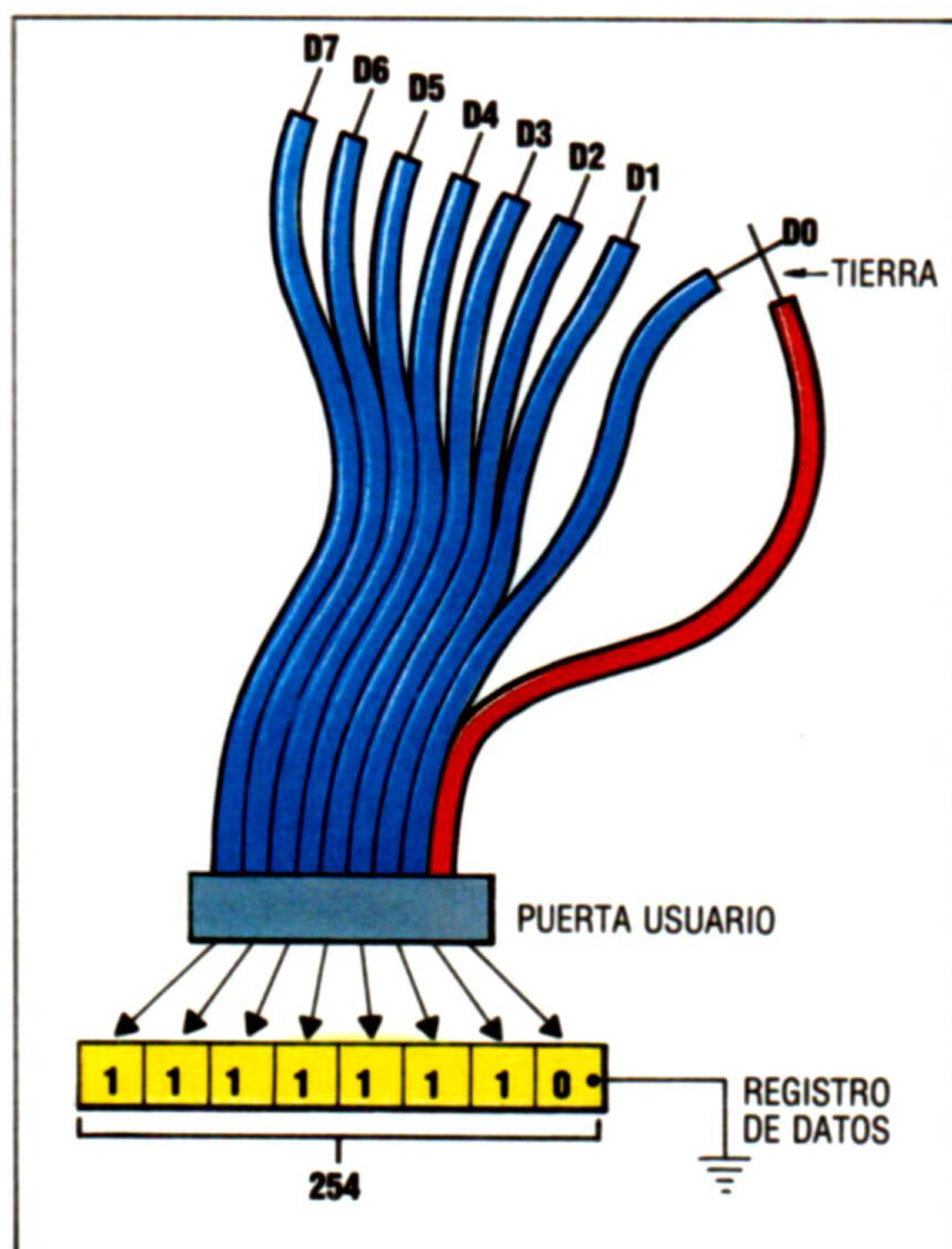




Líneas de conexión
Los cables de la puerta para el usuario del BBC Micro y el Commodore 64

La ejecución del programa muestra que el contenido normal del registro de datos es 255 (en la pantalla HHHHHHHH), lo que significa que las ocho líneas están activadas y las ocho celdas de REGDAT están en el nivel de voltaje +5. Si ahora enchufa un cable en la puerta para el usuario, puede utilizarlo para cambiar los niveles de voltaje en las líneas y, por lo tanto, cambiar el contenido numérico de REGDAT.

Hemos cableado 10 líneas a la puerta para el usuario: ocho de ellas son líneas de datos (una línea para cada bit de REGDAT) y las otras proporcionan una "toma de tierra" al sistema. Hacer que una línea de datos toque una línea de tierra reducirá el nivel del voltaje de la línea de datos a cero, cambiando por consiguiente los niveles de voltaje de REGDAT. Si se conecta a tierra D0, por ejemplo, mientras se está ejecutando el programa, verá que ahora REGDAT contiene 254 (reflejado en la pantalla como HHHHHHHE), lo que significa que la línea menos significativa está en voltaje *tierra* (0 v), mientras que las otras están *altas* (+5 v). Quizá quiera probar distintas combinaciones de líneas para comprobar que, mediante este procedimiento, puede hacer que REGDAT contenga cualquier número entre cero y 255.



Escribiendo software

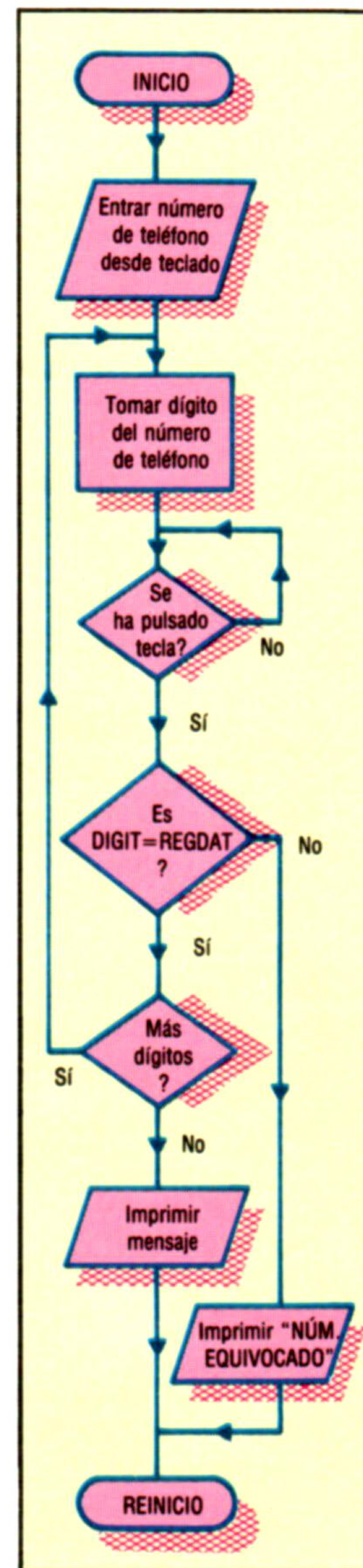
La esencia de la toma de decisiones del ordenador es la comprobación de valores numéricos o condiciones y la bifurcación según el resultado obtenido en la comparación. Por lo tanto, es una cuestión sencilla diseñar software que pueda monitorizar actividades físicas a través de la puerta para el usuario. Mediante la comprobación del valor del registro de datos y emprendiendo la acción apropiada, el ordenador puede responder a los cambios externos. Un ejemplo sencillo sería el de diseñar un programa que compruebe si un número de teléfono se ha marcado correctamente. El "marcado" se puede efectuar conectando a tierra ciertas líneas de datos conectadas a la puerta para el usuario, con el fin de producir en el registro de datos el valor numérico correcto para cada uno de los dígitos del número de teléfono. A causa de las dificultades que entraña conectar a tierra simultáneamente varias líneas de datos, el programa espera que el usuario pulse una tecla del teclado antes de analizar el contenido del registro de datos. La lógica del programa se refleja en el diagrama de flujo que lo acompaña.

```
100 REM ************** C64 ***************
101 REM*          NUMEROS DE TELEFONO          *
102 REM ************** C64 ***************
120 :
130 REGDAT=56577:DDR=56579
140 POKE DDR,0:REM=ENTRADA SOLAMENTE
150 :
160 INPUT"NUMERO DE TELEFONO";NT$
170 :
180 FOR K=1 TO LEN(NT$)
190 DG$=MID$(NT$,K,1): REM TOMAR DIGITO
200 IF DG$ <> " "THEN GOSUB 500
210 NEXT K
250 :
300 PRINT"   ________ LO SIENTO__________ "
310 PRINT" NO HAY NADIE EN EL ";NT$
320 PRINT" POR FAVOR LLAME MAS TARDE"
350 PRINT:PRINT:RUN
400 :
500 REM** S/R CONVERSION Y COMPROBACION**
550 DG=VAL(DG$)
600 PRINT"PONER DIGITO EN LAS LINEAS"
650 PRINT"        Y PULSAR CUALQUIER TECLA"
700 GET GT$:IF GT$="" THEN 700
750 PE=PEEK(REGDAT):IF PE=DG THEN PRINT
    DG" OK":RETURN
800 :
850 PRINT"???NUMERO EQUIVOCADO???"
900 PRINT NT$"<>"LEFT$(NT$,K-1);PE
950 PRINT"???VUELVA A PROBAR???"
999 PRINT:PRINT:RUN
```

Para el BBC, introduzca estas modificaciones:

```
130 REGDAT=&FE60:DDR=&FE62
140 ?DDR=0
700 A=GET
750 PE=?(REGDAT):IF PE=DG THEN
    PRINT DG" OK":RETURN
```

Hemos examinado aquí la entrada de información desde una fuente externa con el fin de afectar el flujo de control dentro de programas. En el próximo capítulo analizaremos la salida a través de la puerta para el usuario y el diseño de un sistema sencillo de control por ordenador.



Líneas cruzadas
El programa de números de teléfono acepta como entrada un número de teléfono (cualquier formato, cualquier longitud) y verifica sus dígitos, uno por uno, con el contenido de la posición de memoria de la puerta para el usuario. Los espacios en el número introducido se ignoran. El programa aguarda una pulsación de tecla antes de comparar un dígito introducido con el dígito de la puerta para el usuario

Ejercicios de programas

Basándose en el programa de números de teléfono:

1) Escriba un programa que simule la acción de una alarma antirrobo.

2) Escriba un programa para contar el número de impulsos recibidos por la puerta para el usuario en un período dado.

3) Modifique su última respuesta para llevar una cuenta separada para cada una de las ocho líneas de datos de la puerta para el usuario.

4) Escriba un programa que simule la acción de una cerradura con combinación.

5) Escriba un programa para cambiar el color de la pantalla desde la puerta para el usuario.

Magia negra

"Necromancer" (Nigromante) es un juego con espectaculares gráficos y en tres actos, como si se tratara de una obra de teatro

Introducción. Tanto la versión Atari como la del Commodore 64 se cargan fácilmente; el lento arrastre de la indolente unidad de disco del Commodore 64 resulta tolerable debido a que el programa produce una gama de extraños ruidos que suenan algo así como una orquesta electrónica afinando. Los gráficos de los títulos se acompañan con una música de presentación que aprovecha al máximo las posibilidades de sonido de ambas máquinas.

Primer acto. El bosque: La pieza comienza en la "edad de la oscuridad", cuando el "supremo soberano es Tetragorn, el malvado hechicero". El jugador hace el papel de Illuminar, un druida al que se describe como "el defensor de la verdad y protector de la raza humana". Como puede imaginar, ello no es tarea fácil. Comienza la obra y aparece el druida en un espacio abierto oscuro. Una atmósfera de estrellas lo protege de centenares de pequeños ogros que marchan sin descanso a través de la pantalla agitando gigantescos machetes al son de un insistente acompañamiento musical. Con la palanca de mando se controla una diminuta varita mágica que vuela dentro de la pantalla destruyendo a los ogros y obteniendo puntos antes de regresar a su mano.

Colocando la varita en el espacio deseado y pulsando el botón de la palanca de mando se pueden plantar árboles en un intento por crear un bosque que le será de ayuda más adelante en el juego. El jugador debe proteger los árboles de los machetes de los ogros y las arañas de Tetragorn, estando atento a sus fuerzas, que se ven mermadas por la picadura de una araña o reforzadas por la muerte de otra. Después de cinco niveles de juego, termina el primer acto con el ataque de las arañas, que agotan rápidamente las fuerzas del druida. El programa entonces congela la acción, cuenta la cantidad de árboles que el jugador ha logrado hacer crecer y sitúa al druida en el siguiente acto.

Segundo acto. Las criptas: Es aquí donde se incuban las arañas. Hay cinco niveles diferentes, cada uno de los cuales contiene dos capas de cuatro criptas. Dentro de las criptas hay huevos de araña, que se iluminan con distintos colores intermitentes cuando éstas están a punto de salir del cascarón. En pantalla también sale el "cofre de los árboles" (que contiene los árboles que se hicieron crecer en el primer acto). El jugador tiene que utilizar su varita para liberar un árbol y desplazarlo a un punto situado sobre una de las criptas; si es rápido, el árbol echará raíces y se abrirá paso a través del techo de la cripta para destruir los huevos antes de que se incuben.

Existe un peligro adicional representado por las "manos del destino"; éstas se estiran desde el techo hacia abajo y cogen todo cuando encuentran debajo de ellas: ya sea el druida, árboles o signos de interrogación. Estos últimos se utilizan para representar premios misteriosos, y debe obtenerse uno de ellos para poder descender por una escalera hasta el siguiente nivel.

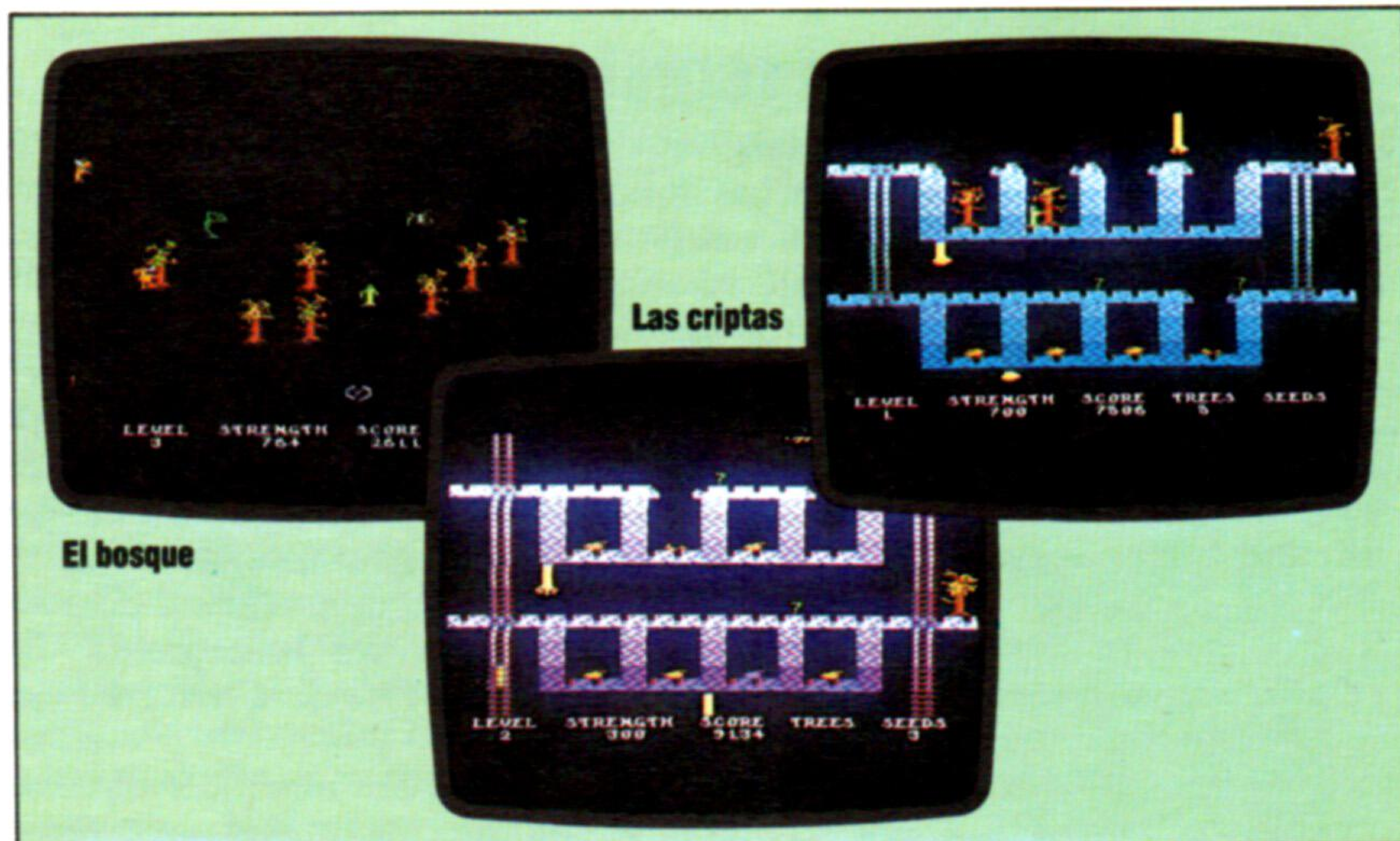
Tercer acto. La guarida del nigromante: El clímax de la obra tiene lugar en un cementerio espectralmente oscuro. Las lápidas marcan las muchas tumbas del Necromancer y se deben destruir para impedir su reencarnación. La varita mágica es lo suficientemente poderosa como para matar a cada reencarnación y las propias lápidas desaparecen si uno consigue colocar al druida sobre ellas. Pero la batalla entre el bien y el mal no es así de sencilla, porque todas las arañas que se escaparon en el segundo acto se transforman en arañas "zombis" y acuden en defensa del Necromancer.

Es una cuestión prioritaria reservar muchísima fuerza para la Guarida, porque aquí el jugador debe valerse de todo su ingenio. Ciertamente el último acto puede resultar muy frustrante, dado que no se puede acceder al mismo independientemente del resto del juego.

La de Atari es la mejor de las dos versiones del juego. La versión para el Commodore es más lenta y contiene gráficos y sonido ligeramente menos espectaculares. Ambas pueden verse deslucidas por palancas de mando de respuesta torpe y lenta. No obstante, dejando de lado estas menudencias, quienes posean cualquiera de las versiones encontrarán que éste es un juego magnífico, aunque caro.

Necromancer: Para el Atari 400/800 (32 K) y el Commodore 64
Editado por: Synapse Software, 5327 Jacuzzi St, Suite 1, Richmond, CA 94804
Autores: Bill Williams (Atari), Scott y Steve Coleman (Commodore 64)
Palancas de mando: Atari/Commodore estándares
Formato: Disco y cassette

Delicia visual
Necromancer combina gráficos sprite con visualizaciones de fondo en alta resolución para producir unas escenas asombrosas en los tres niveles de destreza. Las relucientes copas de los árboles, el rápido vuelo de la varita mágica del druida, el movimiento de las arañas y los ogros y el fuego del nigromante hacen que *Necromancer* sea no sólo un juego emocionante sino también hermoso desde el punto de vista estético



Liz Heaney

Una lengua muy culta

Ésta es la primera lección sobre el lenguaje assembly del procesador 6809. Este lenguaje lo incorporan micros como el Dragon y el Tandy Color

Pueden resumirse en tres los componentes de un microprocesador: los registros, que son partes de la memoria interna del procesador; la ALU (*arithmetic and logic unit*: unidad arimético lógica), verdadera "fábrica" de hacer operaciones con los datos que están almacenados en la memoria; y una unidad de control que se encarga de que todo proceda según el orden previsto y en el momento adecuado.

En su más elemental grado de operatividad, el microprocesador sólo responde a señales de voltaje procedentes de sus conexiones con el exterior y causantes de cambios internos (el contenido de los registros) o de ulteriores envíos y recepciones de señales. Si representamos por un 1 la presencia de voltaje y por un 0 la ausencia de éste, entonces podemos dar valores numéricos binarios a estas señales que van y vienen por la memoria y por todo el procesador. Posteriormente se puede programar el procesador numerando las instrucciones según un determinado orden. Así pues, en el más ínfimo nivel de programación, es imprescindible pensar en términos binarios (o hexadecimales). Ello requiere conocer el efecto de cada número, o código de instrucción en el procesador.

Un procesador de ocho bits como el 6809 puede enviar y recibir números binarios de ocho bits, lo que equivale a una gama de números decimales entre el 0 y el 255. Muchos de los números sirven para indicar las direcciones de las posiciones de memoria, las cuales suelen representarse en la mayoría de estos procesadores de ocho bits con números de 16 bits (por lo que podemos numerar posiciones de memoria entre el 0 y el 65535). Es claro que a la hora de tratar estas direcciones de 16 bits, el procesador lo hace de ocho en ocho bits.

Los registros del 6809

Los registros en un procesador pueden adquirir muchas formas según sus funciones. Algunos quedan reservados para el uso interno del procesador, inaccesibles para el programador. Pero cuatro de ellos serán utilizados por éste muy extensamente.

El más empleado es el *acumulador*. La mayoría de los datos se almacenan en él y en él se manipulan. Por ejemplo, la función realizada por una instrucción en assembly como AND es la de sumar el contenido de una determinada posición de memoria al contenido del acumulador. De esta manera "se acumula" un nuevo valor en este registro.

El *registro índice* sirve para modificar direcciones, lo que permite fáciles recorridos por tablas y listas de datos. Cuando una instrucción se refiere a una posición de memoria por el método del *direccionamiento indexado*, el contenido de este registro va a agregarse al de la dirección dada, con lo que se obtiene la *dirección efectiva* de los datos que necesitamos. Para recorrer una tabla de datos, basta con indicar la *dirección de base* (o sea, la del primer valor de la tabla) e irle añadiendo el contenido del registro. Tales registros índice suelen ser de 16 bits, debido a que los valores que en ellos se almacenan son normalmente direcciones.

El *índice de la pila* es el registro que señala la posición de la parte superior de la pila, para almacenar y tomar datos de ella de la manera más rápida. La pila se emplea para ocasiones en que se necesita preservar el contenido interno del procesador (p. ej., al llamar a una subrutina) de modo que no se pierda y pueda recuperarse más tarde. El contenido de algunos, si no todos, los registros puede "meterse" *(push)* en la pila y "sacarse" *(pop)* fuera más tarde, cuando el programa principal recupera el control. El índice de la pila se limita a facilitar al procesador la posición que ocupa el último valor introducido en la pila y dónde se puede colocar o extraer el valor siguiente. También este registro, al tratar con direcciones de memoria, es por lo general de 16 bits.

El cuarto registro es importantísimo, aun cuando su función sea automática casi siempre. Es el *contador del programa*, destinado a guardar la dirección del programa donde se halla la siguiente instrucción a ejecutar. El procesador accede a la posición que le indicó el registro, toma el contenido de esta posición, interpreta su significado y actúa. El contador del programa suele incrementarse automáticamente una vez ejecutada la instrucción, lo que permite tomar las instrucciones una a una y ordenadamente. Si se altera el contenido del contador (porque se almacena un nuevo valor en él, o se le suma o resta un cierto valor) se cambia el flujo del programa. Es decir, funciona como una instrucción GOTO, conocida a este nivel como un "salto" (JMP; inglés: *jump*) en caso de colocarse una dirección completamente nueva, o como una "bifurcación" (BRA; inglés: *branch*) si sólo se alteró con una operación aritmética la dirección que contenía.

Hay todavía un quinto registro, pero no opera del mismo modo que los anteriores. Es el registro de *código de condición*, al que se ha de imaginar más bien como un grupo de bits individuales, cada uno de los cuales representa alguna característica del estado del procesador. Por ejemplo, uno de estos bits indica si el número resultante de una operación realizada dentro de alguno de los registros es cero. Así, es posible recorrer una tabla de valores



Los populares
Las dos máquinas más populares del 6809 son microordenadores: el Dragon de 32 K y 64 K, y el Tandy Color. Existen además numerosos sistemas desarrollados funcionando en universidades y escuelas técnicas europeas. El Tandy Color y los dos modelos Dragon son internamente muy parecidos, y aunque la Dragon Data ha abandonado el mercado, la Tandy aceptó el compromiso de proveer de software y ayuda técnica a los usuarios de Dragon

cargando el número total de valores en un registro y restándole uno al total cada vez que tratemos con un nuevo valor. Cuando el contador alcanza el cero, el bit del código de condición avisa al procesador que no hay más cantidades en la tabla y que puede pasar a otra instrucción. Esto nos permite realizar selecciones (las sentencias IF) y bucles (tipo FOR...NEXT, WHILE...WEND, REPEAT...UNTIL).

Muchos procesadores poseen la *página cero*, integrada normalmente por las primeras 256 posiciones de memoria (en hexadecimal, de la 0000 a la 00FF). Se puede acceder a los valores en ella almacenados con una dirección de ocho bits, lo que hace a la instrucción más corta y más rápida. El procesador 6809 generaliza este concepto incorporando el registro *página directa* de ocho bits, que se encarga de facilitar los ocho bits que faltan para que la dirección sea completa, al hacer referencia a la página cero. Si se cambia el valor contenido en este registro, la página cero puede colocarse en cualquier lugar de la memoria, incluso se puede disponer de más de una página cero.

Un programa en código máquina se compone de una serie de instrucciones entremezcladas con datos y direcciones. Hay personas que pueden programar directamente manejando los respectivos valores numéricos, pero la mayoría de nosotros encuentra esto muy difícil. El lenguaje assembly de un procesador nos permite escribir programas en lenguaje máquina por medio de representaciones mnemotécnicas (para el caso de las instrucciones) y etiquetas (para direcciones y datos). Así, por ejemplo, si queremos cargar en el acumulador los datos de una determinada posición de memoria. Podemos escribir:

```
STORE FCB 0
```

para que quede reservado un sitio en la memoria al que podamos referirnos por STORE, donde colocamos temporalmente un cero. FCB no es realmente una instrucción, sino una directiva que indica al programa traductor del assembly a código máquina que tiene que sustituir una dirección determinada siempre que encuentre la palabra STORE. Más tarde, cuando el programa desee cargar en el acumulador el valor almacenado de ese modo, se podrá utilizar la instrucción:

```
LDA STORE
```

que tomará el valor almacenado en la posición de memoria referenciada por STORE y lo cargará en el acumulador.

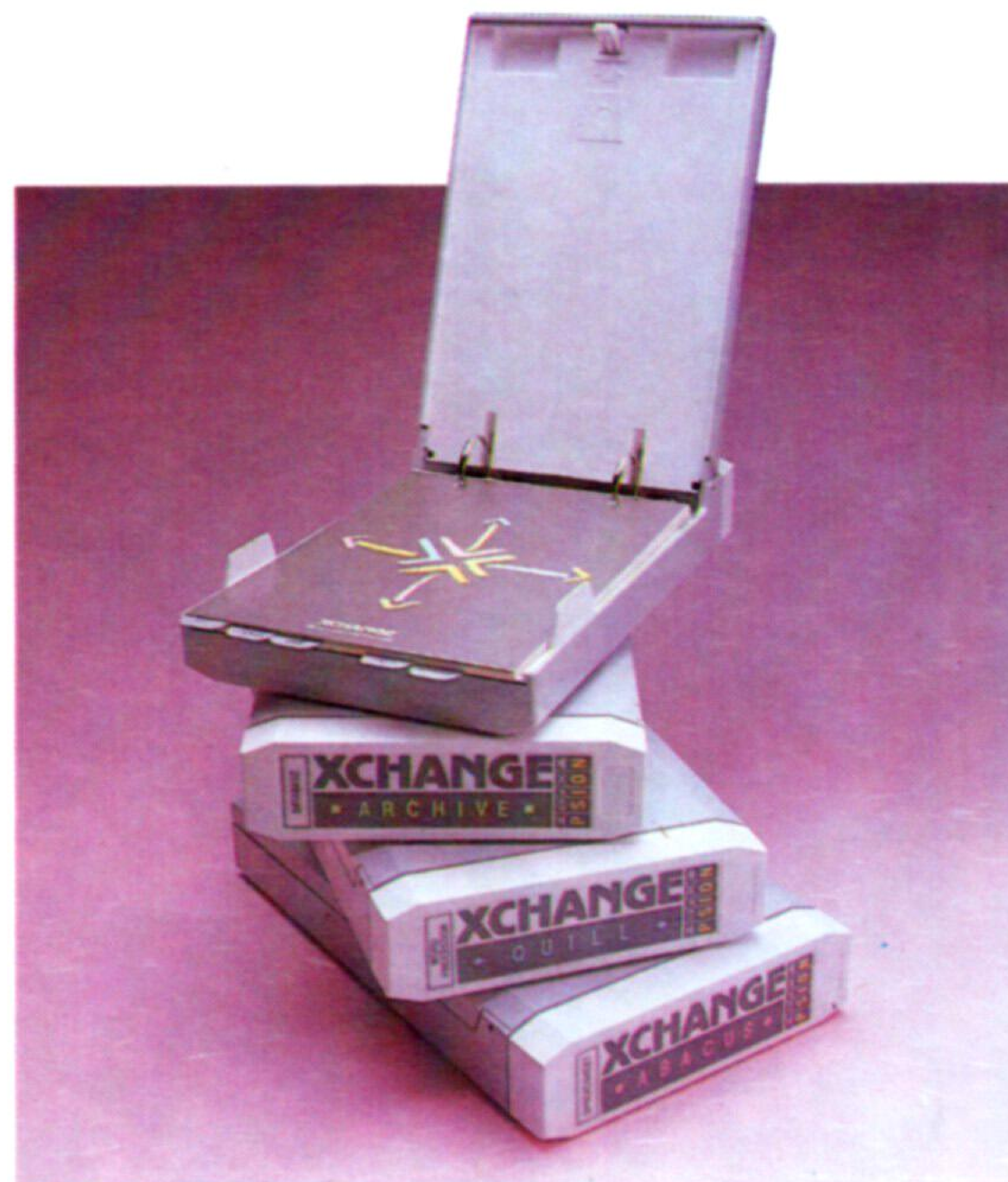
El lenguaje assembly de los programas debe ser traducido previamente a su ejecución, para lo cual se utiliza un programa específico llamado *ensamblador*. No tiene éste por qué ser complicado, pues se trata de una relación biunívoca, de uno a uno, entre las sentencias del lenguaje assembly y sus equivalentes en lenguaje máquina. Todo lo que hay que hacer es practicar la sustitución con cuidado para determinar qué valores o direcciones corresponden a qué nombres.

En la próxima lección estudiaremos la estructura interna del procesador 6809 con mayor detenimiento, iniciando ya el análisis de las instrucciones que tendremos a nuestra disposición.

Nuevos horizontes

Juego integrado
Xchange de Psion es un juego de paquetes de software de gestión integrados basado en el software creado para el Sinclair QL. *Xchange* incluye el procesador de textos Quill, el administrador de base de datos Archive, el planificador financiero Abacus y el sistema Easel para gráficos de gestión. Los cuatro paquetes se pueden adquirir juntos o por separado. *Xchange* está a la venta para el IBM PC y PC XT, ACT Apricot y Apricot XI, y Sirius I. Se están planeando versiones adicionales para el Apple Macintosh y el Digital Rainbow

Recientemente Psion ha diversificado su producción, introduciéndose en el campo del software de gestión



Ian McKinnell

Psion fue fundada en 1981 por David Potter, catedrático del Imperial College de Londres. El primer golpe de marketing de la empresa fue un grupo de cuatro paquetes para el recientemente lanzado ZX81: *Flight simulator* (simulador de vuelo), *Backgammon, Vu-Calc* y *Vu-File*, todos los cuales los escribieron Charles Davies y Colly Myers. Esta pequeña gama de programas de calidad (un paquete de simulaciones, un juego, una hoja electrónica y una base de datos) establecieron inmediatamente el nombre de la empresa. En 1982, cuando Sinclair Research encargó un paquete para demostrar el poderío del Spectrum, no fue ninguna sorpresa que la firma de software elegida para desarrollar la cassette fuera Psion.

La popularidad de las máquinas Sinclair ha abierto un enorme mercado para el software de Psion. La empresa ha tenido algunos éxitos notables: la venta combinada de las versiones del Flight Simulator para el ZX81 y el Spectrum, por ejemplo, ha sobrepadado los 500 000 ejemplares. Las estimaciones relativas al total de las ventas mundiales de cassettes Psion han superado la cota de los tres millones. Y con su reciente anuncio de la gama de software "Xchange", Psion ha dado una primera muestra de su deseo de diversificarse, en este caso haciendo un esfuerzo por conseguir parte del mercado de software de gestión.

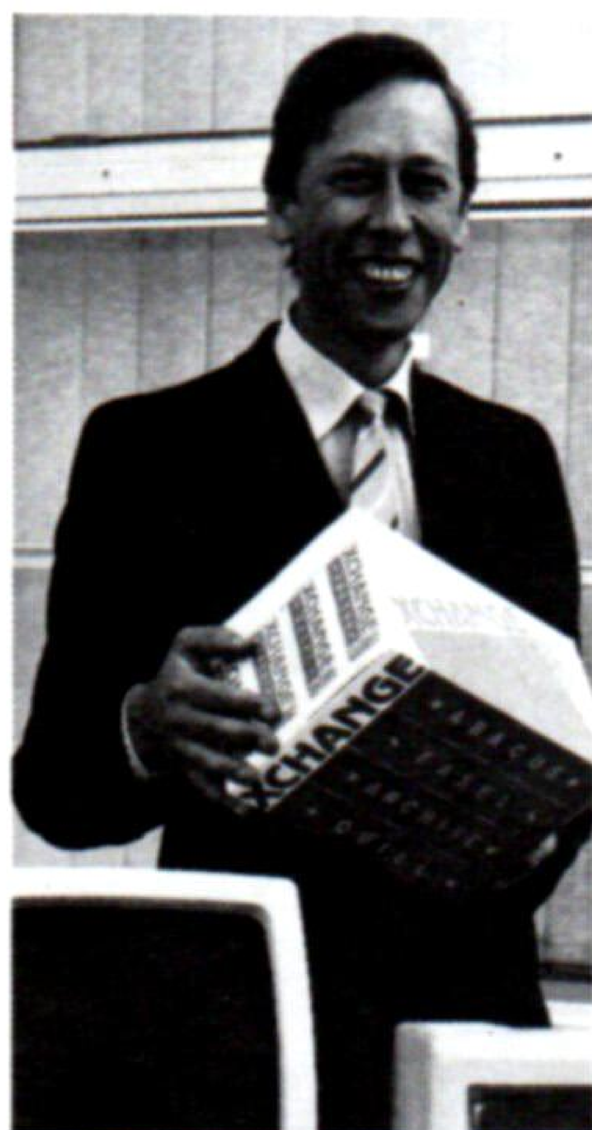
Psion siempre ha sido una innovadora. Abrió el camino para desarrollar la técnica del software "de compilación cruzada" para ordenadores personales, proceso en virtud del cual se desarrolla un programa en una máquina para utilizarlo en otra. El Horizons, paquete de introducción para el Spectrum, se escribió en un Tandy TRS-80. En la actualidad la empresa utiliza dos miniordenadores VAX 750 para todo su software (véase p. 861). Fue en estas máquinas en las que se escribió el juego de programas del QL: Abacus (una hoja electrónica), Archive (una base de datos), Easel (un programa para gráficos) y Quill (un procesador de textos). Psion Support ha organizado QLUB, un servicio de consulta para usuarios del QL que garantiza una respuesta por escrito a cualquier consulta en 48 horas.

Existen planes ya avanzados para numerosos programas de gestión para los ordenadores IBM PC, Apricot, Sirius, DEC Rainbow y Macintosh. Al igual que el juego de programas del QL, la gama de programas *Xchange* configura una hoja electrónica, una base de datos, un programa para gráficos y un procesador de textos. Lo que diferencia a estos paquetes de otros paquetes de gestión, según Psion, es la sencillez con la que se puede transferir datos entre ellos.

Otro de los desarrollos de Psion que ha sido objeto de muchísima atención es el Organiser (véase p. 921). La aparición de un ordenador de bolsillo proveniente de una empresa que hasta entonces se había considerado como de software constituyó una sorpresa para muchas personas. Un portavoz de la firma, Robin Kinnear, enfoca este desarrollo desde otra perspectiva: "La clave es que Psion *es* una empresa de software para microordenadores. Nosotros pensamos que el Organiser era una idea muy atractiva desde el punto de vista del *empaquetamiento* de software y observamos a nuestro alrededor y descubrimos que no existía nada parecido. De modo que Psion decidió hacerse su propio hardware. El desarrollo estuvo en gran parte determinado por el software".

En la actualidad existen sólo tres paquetes de aplicaciones para el Organiser (ciencias, matemáticas y finanzas), junto con los paquetes de RAM (o *datapacks*, como se llaman) de 8 Kbytes y 16 Kbytes. Psion está considerando la posibilidad de agregar otros paquetes de programas; asimismo, diversas personas que desean escribir software para la máquina se han puesto en contacto con la empresa.

La expansión de mercado es otra de las prioridades de la empresa en estos momentos: recientemente se han establecido subsidiarias en Estados Unidos y Sudáfrica, y se han firmado contratos para la distribución de los productos de Psion en toda Europa. Además, Sinclair Research ha lanzado una gran campaña de ventas en Europa Oriental, empezando con la exportación de 400 ZX81 a Checoslovaquia. Psion, que ya se ha tomado cierto trabajo en la producción de versiones de su software en lenguas foráneas, está segura de seguir avanzando.



Catedrático y empresario
David Potter es el fundador y principal accionista de Psion. Anteriormente académico especializado en Computational Physics en el Imperial College de Londres y en la Universidad de California, fundó Psion como Potter Scientific Investments


mi computer
Editorial Delta, S.A.



10050

9 788485 822836

